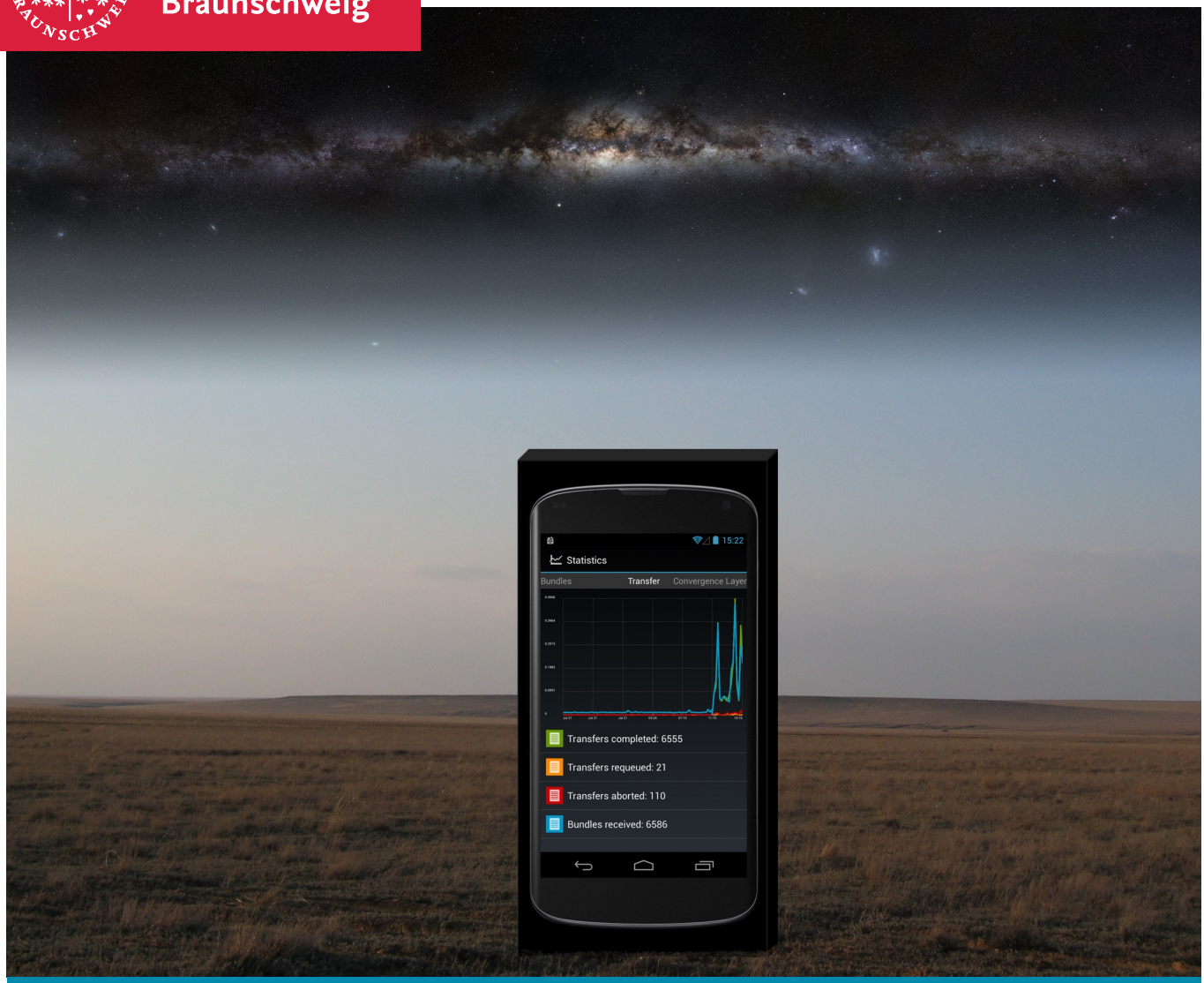
Institute of Operating Systems
and Computer Networks

Technische
Universität
Braunschweig



Scaling Up Delay Tolerant Networking

Sebastian Schildt

Scaling Up Delay Tolerant Networking

Von der
Carl-Friedrich-Gauß-Fakultät
der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung des Grades eines
Doktoringenieurs (Dr.-Ing.)

genehmigte Dissertation

von
Sebastian Schildt
geboren am 29.11.1981
in Stade

Eingereicht am:	28.05.2015
Disputation am:	01.12.2015
1. Referent:	Prof. Dr.-Ing. Lars Wolf
2. Referent:	Prof. Dr.-Ing. Jörg Ott

2016

Contents

Abstract

Delay Tolerant Networks (DTNs) introduce a networking paradigm based on store, carry and forward. This makes DTNs ideal for situations where nodes experience intermittent connectivity due to movement, less than ideal infrastructure, sparse networks or other challenging environmental conditions. Standardization efforts focused around the Bundle Protocol (BP) (RFC 5050) aim to provide a generic set of protocols and technologies to build DTNs. The BP can be layered on top of various protocols, including standard IP networks. This makes the BP a suitable technology for fringe networks with intermittent access to the Internet such as networks of mobile smartphone users, VANETs or various types of Wireless Sensor Networks (WSNs). To connect and interconnect these fringe DTN networks with the Internet, the BP should be used as end-to-end transport.

However, there are several challenges when trying to apply the BP to the Internet as a whole that are tackled in this thesis: There is no DTN routing mechanism that can work in Internet-scale networks. Similarly, available discovery mechanisms for opportunistic contacts do not scale to the Internet. This work presents a solution offering pull-based name resolution that is able to represent the flat unstructured BP namespace in a distributed data structure and leaves routing through the Internet to the underlying IP layer. A second challenge is the large amount of data stored by DTN nodes in large-scale applications. Reconciling two large sets of data during an opportunistic contact without any previous state in a space efficient manner is a non-trivial problem. This thesis will present a very robust solution that is almost as efficient as Bloom filters while being able to avoid false positives that would prevent full reconciliation of the sets. Lastly, when designing networks that are based on agents willing to carry information, incentives are an important factor. This thesis proposes a financially sustainable system to incentive users to participate in a DTN with their private smartphones. A user study is conducted to get a lead on the main motivational factors that let people participate in a DTN. The study gives some insight under what conditions relying on continuous motivation and cooperation from private users is a reasonable assumption when designing a DTN.

Kurzfassung

Delay Tolerant Networks (DTNs) sind ein Konzept für Netzwerke, das auf der Idee beruht, Datenpakete bei Bedarf längere Zeit zu speichern und vor der Weiterleitung an einen anderen Knoten physikalisch zu transportieren. Diese Vorgehensweise erlaubt den Einsatz von DTNs in Netzen, die häufige Unterbrechungen aufweisen. Mit dem Bundle Protocol (BP) (RFC 5050) wird ein Satz von Standardprotokollen für DTNs entwickelt. Das BP kann über einer Vielzahl darunterliegender Netzwerkschichten, wie zum Beispiel IP, eingesetzt werden. Daher ist das BP gut für den Einsatz an der Peripherie eines Netzes geeignet, wo mit häufigen Unterbrechungen der Konnektivität zu rechnen ist. Beispiele für derartige Netze sind mobile User mit Smarphones, VANETs oder Wireless Sensor Networks (WSNs). Um diese Netze mit dem Internet zu verbinden, sollte das BP als Ende-zu-Ende Protokoll eingesetzt werden.

Bei der Verwendung des BP ergeben sich einige Herausforderungen, die es zu überwinden gilt, wenn man das BP im Internet einsetzen möchte: Es existiert kein DTN Routingverfahren, das skalierbar genug ist um im Internet eingesetzt zu werden. Das Gleiche trifft auf verfügbare Discovery Mechanismen für opportunistische Netze zu. In dieser Arbeit wird ein verteilter, reaktiver Mechanismus zur Namensauflösung im DTN vorgestellt, der den flachen, unstrukturierten Namensraum des BP abbilden kann und es erlaubt das Routing komplett der IP Schicht zu überlassen. Eine weitere Herausforderung ist die große Menge an Nachrichten, die Knoten puffern müssen. Die effiziente Synchronisierung von zwei Datensets während eines opportunistischen Kontaktes, ohne Zustandsinformationen, ist ein komplexes Problem. Diese Arbeit schlägt einen robusten Algorithmus vor, der die Effizienz eines Bloom Filters hat, dabei jedoch die False Positives vermeidet, die normalerweise eine komplette Synchronisation verhindern würden.

Ein DTN basiert darauf, dass Teilnehmer Daten puffern und transportieren. Wenn diese Teilnehmer z.B. private User mit Smarpthones sind, ist es essentiell diese Benutzer zu einer dauerhaften Teilnahme am Netzwerk zu motivieren. In dieser Arbeit wird ein finanziell tragfähiges System entwickelt, welches Benutzer für eine Teilnahme am DTN belohnt. Eine Benutzerstudie wurde durchgeführt, um herauszufinden, welche Faktoren Benutzer motivieren und unter welchen Umständen davon auszugehen ist, dass Benutzer dauerhaft in einem DTN kooperieren und Resourcen zur Verfügung stellen.

1 Introduction

Delay Tolerant Network (DTN) is a concept for networks that suffer frequent disconnections and large communication delays. A large, and one of the first, application areas for DTNs are space missions. Long delays due to the speed of light prevented standard protocols such as TCP to be effectively used over long distances. Disruptions, when a line-of-sight communication link is broken or due to the movement of spacecrafts and celestial objects, make lots of best practices usually applied in classic networks largely ineffective. Therefore, in the 90ies work started to develop and implement Interplanetary Networks (IPNs), a DTN concept optimized for space applications. Similar communication challenges were encountered by the Wireless Sensor Network (WSN) community, when deploying sparse networks of mobile nodes [1]. It became clear early on, that classic networking paradigms and protocols such as can be found in the Internet are not applicable for DTN.

In a DTN, nodes implement a Store-Carry-And-Forward behavior: Information is not immediately transferred, but instead network nodes are extended with a *storage*, that allows them to keep hold of information until such a time when a new suitable contact comes in range. This is different from buffers in classic network architectures. In a DTN the storage is expected to hold and manage a considerable amount of data for prolonged times. This allows information to be delayed much longer than in classic networks. But by incorporating long-term storage into a network you gain an additional dimension for routing: Instead of routing through a network's topology in space, it is now possible to route through the space-time topology in a network, using network capacities inaccessible to classic networking approaches.

While the first DTN ideas and implementations were developed for IPNs, people began to see that DTNs are a powerful concept in other kinds of networks: There can be a monetary and efficiency benefits in moving applications from low bandwidth cellular networks to high bandwidth short range communications such as Wi-Fi, exploiting mobility of network nodes. With the nearly ubiquitous availability of personal communication and computation devices such as smartphones or tablets a there is clear opportunity to enable DTN communication independent from infrastructure between personal devices. Such networks of smartphones have been proposed for novel P2P applications or to increase the range and quality of service at the fringes of a cellular network, or to offload data from a cellular network.

When using DTNs in this non-IPN application fields there are some additional challenges. In contrast to IPNs these networks are a lot less deterministic, which complicates routing and reliability. In many useful application scenarios for DTNs, when the DTN is operating at the fringe of some network where service quality is bad, or when it operates under some

conditions where no infrastructure is available, it is a reasonable assumption that the applications will include some sort of gateway to the Internet. Probably at some point in time, collected data from a DTN application should be sent to some back-end service or new data needs to be injected into a DTN network from a central instance. Instead of seeing this as heterogeneous applications with an Internet and a DTN part that need to be bridged together by some - probably application specific - gateway we propose seeing them as convergent applications: We propose using the DTN technologies end-to-end, even if you hardly expect any Delays and Disruptions at all and 99% of your application can rely on continuous connectivity. There are good reasons for this: Even though the Bundle Protocol (BP), which we will focus on in this thesis, is designed to handle disruptions it can do everything that TCP/IP can: BP can be seen as a superset of TCP/IP offering delay and disruption tolerance as an added bonus. Available BP implementations are very efficient and their overhead and performance compares favorably to other approaches. By using the BP end-to-end there is no breach of networking paradigms in your application, there is no need to develop application-specific gateways to gain DTN capabilities. Additionally, the application can work transparently across different communication technologies such as a classic TCP/IP based Wi-Fi network and a ZigBee-based sensor network.

Although the BP is already up to the task to be used end-to-end across the Internet there are some challenges which so-far have precluded the adoption of BP in convergent large-scale Internet-DTN applications:

1. Existing DTN routing mechanisms either route a packet directly to a neighbor or use some flooding scheme. Obviously, flooding is not a feasible way to perform routing or neighbor discovery in the Internet.

2. When adopting DTNs as the network layer for applications, and considering the possible scale of Internet connected DTN applications each DTN node has a potentially large number of bundles in its local storage. When two such nodes met, it is to be expected that they use that opportunity to exchange some of their stored bundles. The routing protocols will decide, which bundles to exchange in what order. Independent of the exact workings of the employed routing mechanism information about the difference set is needed: Which bundles are available on one node, but not the other. Determining the difference set is not trivial for large collections, as the overhead of exchanging simple lists gets to large.

3. In cases of offloading data to a mobile user's private equipment or in general forwarding application's data through private devices, the question of incentives arises: Selfish users will hurt the network's overalls performance. But why should a user offer storage and energy, which will limit the usable time for his device, without any direct compensation? The challenge is enticing users in an economically viable way to continuously participate in DTN.

1.1. Outline

In Chapter 2 we start by introducing the concept of a DTN and looking at the BP. In Chapter 3 we look at the challenges and current deficiencies of currently available DTN technologies when it comes to routing and neighbor discovery in the Internet or other very large networks. We propose several new technical solutions to overcome this challenge and we will show how existing Internet infrastructure can be used to strengthen Internet-based DTNs. Since an Internet-scale DTN will need to deal with a substantial amount of traffic, nodes need to store a large amount of data. Chapter 4 examines how we can efficiently and reliably synchronize the data stored on two nodes. Since a large part of the Internet today is accessed by mobile devices, which especially profit from DTN technologies when moving through areas with poor network connectivity we look at the problem of motivating users to invest their devices' capabilities into a DTN in Chapter 5. Finally, in Chapter 6 we will conclude the work presented in this thesis, reiterate the main contributions of this thesis and discuss how the applications and usage of DTNs will likely evolve in the future.

2 DTN Basics

In this Section we will introduce the DTN idea and look at the BP which is a specified set of protocols to implement DTNs.

2.1. Delay Tolerant Networking

DTN technologies are designed for networks, in which the underlying assumptions of the Internet Protocol (IP) cannot be guaranteed. DTN approaches replace the end-to-end semantics of common protocols such as IP with a hop-by-hop store, carry and forward architecture [2]. While early application-specific implementations of a store, carry and forward architecture can be found for applications where nodes might see each other only occasionally [1], the area of IPN was the first to develop common requirements and protocols for DTN. Today DTN technologies have also been applied for other kinds of networks [3] that provide challenges such as the high mobility in vehicular networks. A DTN between devices can be used when infrastructure is temporarily not available. It can also be used when a cellular network is avoided for cost reasons, or when direct, high bandwidth transmissions between devices are more effective than using the infrastructure [4].

The hop-by-hop approach of DTNs moves the point of retransmissions towards the destination which reduces the network load. If a packet is dropped or gets corrupted, it can be stored and retransmitted by the last hop the packet reached and not by the source. In classical IP networks with end-to-end semantics usually the source is responsible for a packet. If an acknowledgment for a delivered packet is not received, the source node is responsible for sending the data again. Nevertheless, a DTN can optionally support end-to-end acknowledgments, but due to possibly large delays this is often not practical. Alternatively, to enable reliable transmission DTNs introduce the idea of "Custody Transfer": A bundle can be transported from hop-to-hop and each new hop optionally "takes custody", meaning it is accepting responsibility for the final delivery of the bundle. Since data is transferred in a store, carry and forward way, temporarily unavailable links can be compensated. Data is buffered in the storage of the nodes and will be forwarded whenever a suitable link is available again. In fact, DTNs do not require a continuous path from source to destination to exist at any point in time. Instead, time is merely an additional dimension for routing: There has to exist a path between source and destination throughout the passing of time (before the bundle expires) for the bundle to eventually reach its destination. This provides an opportunity to exploit movement of nodes to transport data towards the destination.

The most widespread technical realization of a DTN network is the Bundle Protocol (BP) [5], which we will introduce the BP in Section 2.2.

Company	URL
Amazon Inc.	`http://aws.amazon.com/importexport/`
Backblaze Inc.	`https://www.backblaze.com`
Code 42 Software, Inc.	`https://www.code42.com/crashplan/`
Carbonite, Inc.	`http://www.carbonite.com/`
EMC Corporation	`http://mozy.com/`

Table 2.1.: Companies mailing or receiving data on hard disk (2015)

2.1.1. General DTN Use-Cases

As we have already mentioned, for IPNs DTN concepts are a necessity to enable communication over multiple hops. Outside of IPNs there are several main reasons to adopt DTN technologies. A DTN can be deployed whenever no infrastructure is available. Building infrastructure is comparatively expensive, especially in underdeveloped regions or due to difficult terrain [6, 7]. Even if infrastructure in form of a cellular network is available, its capacity might be limited. If an application demands large bandwidth, a cellular network might not be able to provide it. In this case, offloading data to a DTN increases the achievable bandwidth [4]. Because a DTN can operate with very sparse networks, deploying a DTN can be more cost-effective than paying for access to a cellular network.

Even with evolving networks and increasing coverage a DTN will always have a bandwidth advantage. In general, for wireless standards of the same age, higher bandwidth standards only allow for a short communication range while the longer the range, the more limited the bandwidth becomes. An example is the relatively low bandwidth of cellular networks compared to Wi-Fi technologies.

In the past the increase in available bandwidth has always been coupled with the development of new applications which saturated the available bandwidth. This can be well observed by the increase in Internet traffic [8, 9]. Thus it is not an option to just wait until some specific network achieves total coverage. As of today GSM connectivity is almost ubiquitous in Germany, and can be used for transferring data. However today not even the simplest of web pages are usable with baseline GSM bandwidth of 9.6 kBit/s.

Even with the increasing bandwidth in deployed communication networks, the famous example from Computer Networks 101, “Never underestimate the bandwidth of a station wagon full of tapes hurtling down the highway.” [10], is still true. Randall Munroe from XKCD tried to answer the question “When - if ever- will the bandwidth of the Internet surpass that of FedEx?”[11]. In the hypothetical example of FedEx transporting only data, given current storage technology. In 2012 the Internet traffic averaged 167 Tbit/s while Fedex’ bandwidth using 64 GiB microSD cards is estimated at 177 Pbit/s. Based on current growth rates the Internet would surpass FedEx’s current capacity only by 2040. However, storage density will also improve and thus “[...] the Internet will probably never beat SneakerNet”[11].

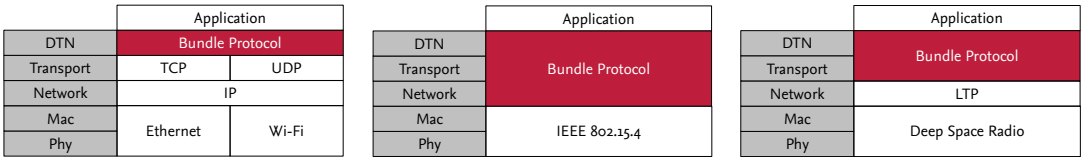
In summary, whenever there is the need for large bandwidths and latency is of less importance, a DTN beats classical networks in price and performance. Even with improving communication technologies this will not change, because shorter links will intrinsically be able to support higher bandwidth, and storage density increases will make sure physically moving data retains its bandwidth advantage. For exactly this reason storage providers offer sending or receiving hard disks as a fast backup seeding or restore mechanism when the customers' Internet access is slow. Table 2.1 shows some companies providing this service. What an efficient DTN protocol stack can do in this case, is making this operation transparent to the software. There would be no difference between using Ethernet to communicate with a server next door, or receiving network data from a hard disk that was flown around the planet and has just been connected.

2.1.2. DTN in the OSI stack

A DTN-capable narrow-waist protocol has huge benefits for current and future communication challenges. While DTN is a concept, this work focuses primarily on the BP [5], which is the most widespread specification and de-facto standard for DTN networks. Basically the BP offers all features typically expected from a network layer and a lot of transport layer functionalities comparable to TCP. However, when we talk about the BP as narrow-waist, this does not imply to replace IP with the BP. This would be an unrealistic assumption and, as we will see, not always the wisest technical choice. In fact, the idea of a DTN narrow-waist is that many applications would benefit being built on top of a DTN layer as lowest common network denominator instead of IP. However, different from other proposals, suggesting using HTTP in a similar way [12], the BP has the advantage that it not just adds another layer of abstraction, but that it can be applied in a wide range of scenarios without taking any of the classical TCP/IP functionalities away.

One use case are classical Internet applications: These run on normal servers, desktop PCs and mobile computing devices such as tablets or smartphones. Here you want to keep the capabilities of the IP network, which includes efficient world-wide routing. Where applications are benefiting, is when mobile devices on peripheral networks are included. These devices will do a lot of vertical handovers between 3G/4G and Wi-Fi networks and regularly left without any connection for some time. Instead of letting every application deal with this failure modes individually, for a DTN-aware application the BP deals with these challenges. Also, the store, carry and forward principle allows new innovative applications such as data harvesting in a participatory Smart City sensing application. A typical protocol stack for these kinds of applications can be seen in Figure 2.1a. Technically, there are some challenges when applying the BP to the Internet as a whole, which will be discussed in Chapter 3.

Another growing application area of networking are Wireless Sensor Networks (WSNs). The primary design goals for WSN nodes are usually low cost and energy efficiency so they can operate for prolonged times (maybe years) from a battery. To achieve this, low-power, low-bandwidth communication standards such as IEEE 802.15.4 and small micro controllers with only a few kiB of RAM are used. In these cases it is possible, to use the BP



(a) Internet Applications (b) WSNs and IoT (c) IPN

Figure 2.1.: DTN stack configurations for different application areas

on top of the MAC layer (which is usually implemented mostly in hardware), achieving performance metrics comparable to those of small IP implementations on the same platform [13]. Recently there are some strong forces pushing for convergence between WSNs and the Internet under the banner of the "Internet of Things". An exemplary DTN stack for these devices is shown in Figure 2.1b. By using the BP on both sides, the gap between those widely different approaches to networking can be bridged.

Last but not least, for Interplanetary Networking, the area which served as an incubator for DTN ideas, the BP is usually stacked on top of the Licklider Transmission Protocol (LTP) [14], which has been specifically designed to deal with long-haul links with long round-trip times. The corresponding protocol stack is shown in Figure 2.1c.

In summary, the BP can be used as an efficient, generic way to add DTN capabilities to existing networks as an overlay using standard protocols such as TCP, or it can be put directly on top of a MAC layer. This allows the BP to scale from large-scale servers down to small sensor nodes.

2.2. The Bundle Protocol

The most widespread specification for DTN network is the BP [5]. The BP has been designed by the IETF DTN Research Group (DTNRG)[1]. In 2014 an IETF Working Group (DTNWG) has been formed with the expressed goal of evolving RFC 5050 and eventually putting it on a standard track. Currently the DTNRG and DTNWG coexist, with largely the same stakeholders. The focus of the DTNWG are slight adaptations and streamlining of the BP, while more experimental work should take place in the DTNRG. The BP as defined in RFC 5050 [5] is only one way to realize the DTN concept. However, today it is the most general and widely applied DTN protocol. Non-BP DTNs are mostly application-specific solutions.

Several inter-operable BP-DTN implementations exist. Most notably are DTN2[2], which acted as reference implementation and was originally pushed by the DTNRG. IBR-DTN[3] is a very efficient and fully featured BP stack that among other things also runs on embedded devices [15] or smartphones [16]. Lastly there is ION[4], a BP implementation developed by NASA that focuses on IPN scenarios. It is usable in real-time environments but can only

[1] http://dtnrg.org/
[2] http://sourceforge.net/projects/dtn/
[3] https://github.com/ibrdtn/ibrdtn
[4] http://sourceforge.net/projects/ion-dtn/

deal with deterministic contact schedules.

While the challenges discussed in this thesis are relevant to DTNs in general, most the implementations presented in this thesis have been built on top of the BP ecosystem. The BP defines the wire-format for DTN messages. The following gives a short overview of the main building blocks of the BP

Bundles

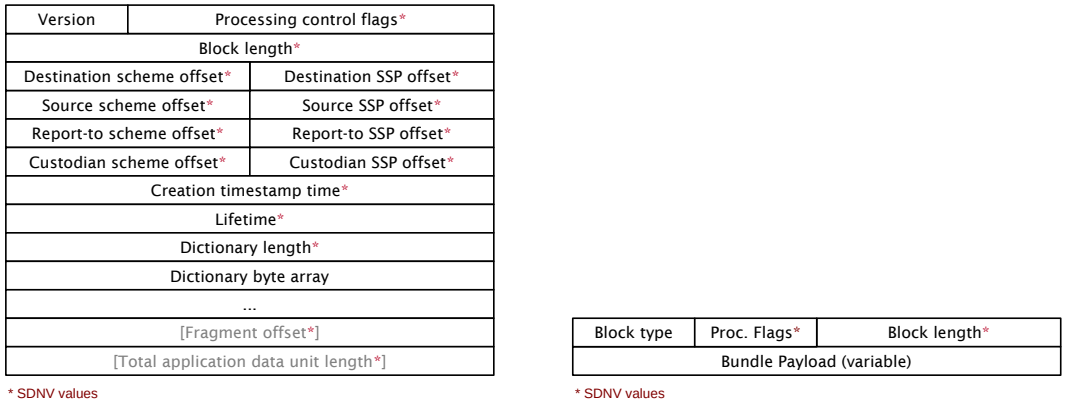
In BP the basic Protocol Data Unit (PDU) is a bundle. Compared to frames or packets of other protocols, bundles can be much larger. As end-to-end delays in a DTN can be high, the idea is to put all data that forms a logical useful block of data into a single bundle. When data is transferred, a BP implementation tries to make sure that nodes always receive complete bundles of data. This aims to avoid the condition that only some segments of a bundle are received, while other segments due to the nature of a DTN might never arrive. What belongs to a single bundle is determined by the application. As a rule of thumb a single bundle should contain enough information to be useful to the receiving application without depending on data from other bundles.

A bundle itself consists of several blocks. Every bundle contains the Primary Bundle Block (PBB) and one or more other blocks. The PBB is depicted in Figure 2.2a. It contains the destination and source addresses of a bundle. Additionally, the BP contains the notions of a "Report-to" and a "Custodian" address. As mentioned before, the BP allows optional acknowledgments or reports that data has been forwarded to a specific peer, not unlike ICMP in IP networks. These reports will be sent to the "Report-to" address. Optionally a bundle can have "Custodian". While a bundle might be replicated to many nodes and at the same time might be deleted by any node under arbitrary conditions such as storage congestion, a "Custodian" takes on special responsibility for a bundle and will make sure the bundle stays alive in the network. All strings, such as the addresses, are stored in a central "dictionary", to save space. Fields such as the addresses just contain pointers to the dictionary, allowing common strings to be used multiple times. However, as of 2015 there are discussions to remove the dictionary concept for the upcoming 5050bis version of the BP, as its space savings do not seem warrant the complicated parsing. While a BP DTN tries to treat bundles as atomic units, there are certain conditions when fragmentation might be desired, therefore the PBB provides the fragment-offset field to identify bundle fragments.

The most common block besides the PBB is the payload block (see Figure 2.2b). Special blocks exist for the BP security extensions or for dealing with relative time. The design of the BP is open, so that applications or implementers can add new custom blocks without violating RFC 5050.

Convergence Layers

The BP itself is transport-layer agnostic. It provides addressing and supports various routing protocols, which have been built on top of the BP. However, a lower level protocol is needed to transport data between two BP nodes. In BP these lower levels are called



(a) Primary Bundle Block (b) Bundle Payload Block

Figure 2.2.: Bundle Protocol blocks

Convergence Layers (CLs). Usually a CL is a very small layer that often implements chunking of data transfers, as most protocols do not support arbitrarily large packets like the bundles. Also, usually some mapping or integration with the addressing scheme of the CLs protocol is needed. For practical reason the most widely implemented CL is the TCP Convergence Layer (TCPCL) [17]. Another common CL is based on UDP [18]. For space applications LTP [14] is often used. The UDPCL is about to be superseded by the newer, more generic Datagram CL [19]. However, using BP on top of TCP/UDP is merely a convenience. As the BP provides a lot of the same functionalities TCP/IP does it can easily work on more low level CLs, such as WSN MAC layers. For example, the BP has been implemented on top of the IEEE 802.15.4 [20] MAC for sensor networks [13]. On the other hand, as we will show in Section 3.6 for some applications it makes sense to transport the BP over existing application level protocols. The DTN paradigm also allows for more unconventional CLs, such as a File CL, that serializes bundles on some storage to be carried around.

EIDs

Addresses in a BP network are called Endpoint Identifiers (EIDs). They are introduced in RFC 4838 [2], which describes the architecture of a BP-based DTN in general. Syntactically EIDs are Uniform Resource Identifiers (URIs) as defined in RFC 3986 [21]. However, the scheme or semantic of an EID URI is not defined. Furthermore, RFC 4838 explicitly states that an EID does not need to be a node identifier. It might designate a group or an "interest" or express any semantic a designer likes. However, for practical purposes, most deployments have adopted simple EIDs such as *dtn://node/application* with addresses in the form of *dtn://mygroup* already considered advanced.

SDNVs

As can be seen in Figure 2.2 numerical fields in the BP are often encoded as Self-Delimiting Numeric Values (SDNVs). SDNVs are defined in RFC 6256 [22]. Basically an SDNV is an arbitrarily sized integer. The Most Significant Bit (MSB) of a byte indicates whether the

integer is continued in the following byte or not. Thus, a value of up to 127 can still be encoded in a single byte, while at the same time there is virtually no conceptual limit on the maximum number that can be expressed. Of course, for practical reasons most implementations limit the SDNV they can process to the range of 64 bit integers.

3 Internet-scale Routing and Naming

3.1. Problem Statement

As we have seen, the Bundle Protocol is an efficient, standardized way to implement a DTN. With the BP's Convergence Layer concept, it can be adapted to all kinds of transport networks. The TCP Convergence Layer [17] is implemented by most BP implementations and is thus a good choice when applying the BP to an IP network such as the Internet. We have argued, that it is advisable to implement a DTN end-to-end. Applications should be built on top of the BP instead of using TCP/IP and dealing with disconnections on the application level.

This implies that the BP needs to work in networks many orders of magnitude larger than in common island-style applications. Compare the number of expected participants of a file sharing application using ad-hoc Wi-Fi with the size of a network, when an application like Facebook uses the BP. According to data published by Facebook as of 2014 [1] the number of monthly active users is 1.3 billion, with 47% of users logging in each day.

This leads to the closely related challenges of discovery, routing and naming that need to be overcome: In the BP every endpoint is identified by its EID, which can be an arbitrary URI. The current approach is to find neighboring nodes by some beaconing/broadcasting mechanism. This obviously does not work in the Internet as a whole. Additionally, despite intuition, beacon-based discovery is also not always a good approach for small-scale ad-hoc applications due to energy constraints. Most DTN multi-hop routing mechanisms that can be applied in opportunistic scenarios are inherently based on flooding. Such mechanisms are not applicable to the Internet as a whole. What is missing is an efficient, resilient way to contact a DTN node in the Internet, when only its EID is known.

Additionally, massively scaling up DTN applications will require more BP-aware hosts and routers in the Internet. However, it is not reasonable to assume, that a huge parallel infrastructure of a "new" and better Internet will be built and operated besides current hard- and software. This is the same challenge many Next Generation Internet approaches face. A more realistic approach is, using as much from the legacy infrastructure as possible, allowing DTN applications to grow organically.

In this chapter we will first look into the problem of discovery and present a set of extensions that enable BP-based DTNs to scale to networks as large as the Internet in

[1] http://www.statisticbrain.com/facebook-statistics/

a cost-effective and efficient way. In Section 3.6 we will present a solution that to some extent can offset the need for BP routers by using the Internet Mail system, enabling more reliable and energy efficient communication for mobile users.

Parts of the work presented in this chapter have already been discussed in [23, 24, 25].

3.2. Making Friends: Discover Other Nodes

In opportunistic DTN scenarios, when there is a communication opportunity between two nodes, the aim is to maximize utilization of the contact. The goal is to transfer as much data as possible during the contact. Before a contact becomes useful, both nodes must be aware of the contact. How easy this is depends on the type of contact. We extend and modify the terminology introduced in RFC4838 [2] to define different types of DTN contacts shown in Figure 3.1:

1. *Persistent Contacts:* Contacts that are always available. An example is a DTN node connected to the Internet hosted inside some data center. This is the equivalent of a server in classical Internet applications. As these nodes are considered to be always online, there is no need for any dynamic discovery mechanism once they are known.

2. *Opportunistic Contacts:* Contacts that are neither scheduled nor predicted. From the system's point of view these contacts are random. Examples are DTN P2P applications using mobile devices. These kinds of DTN applications are sometimes termed Pocket Switched Networks [26].

3. *Scheduled Contacts*: Scheduled contacts are established at a particular time and for a particular duration. Time and duration are known beforehand. There are two types of scheduled contacts:

   a) *Hard-Scheduled Contacts*: Time and duration of a contact are known with a high degree of precision. Examples are classical IPN scenarios. Just as with persistent contacts a discovery mechanism is not necessary.

   b) *Soft-Scheduled Contacts*: Time and duration of contacts are known beforehand, however both time of establishment and duration have significant amounts of uncertainty attributed to them. Examples are DTNs in public transportation systems [27], where a timetable provides some degree of determinism.

4. *Predicted Contacts*: This class is somewhat less sharply defined. This can encompass contacts between scheduled and opportunistic. Depending on the history and some domain specific information future contacts will be predicted. This includes a wide range of contact situations: Soft-scheduled contacts from time-tables can be seen as predicted. Certain statistic properties from opportunistic contacts, such as those used by the PRoPHET routing [28], can also be used to predict contacts. While in the first case a contact can be predicted with a reasonable degree of certainty within a couple of minutes, PRoPHET on the other hand only "predicts" that some node

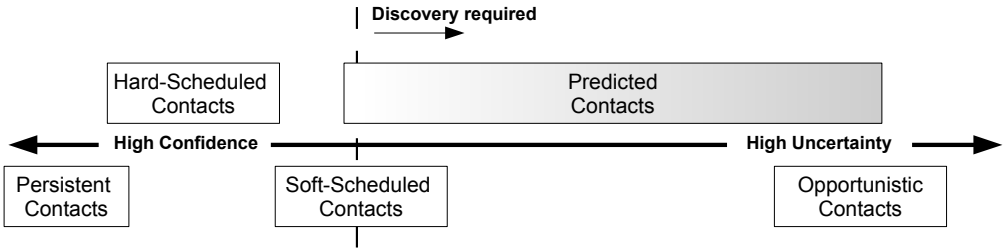


Figure 3.1.: Contact types in a DTN

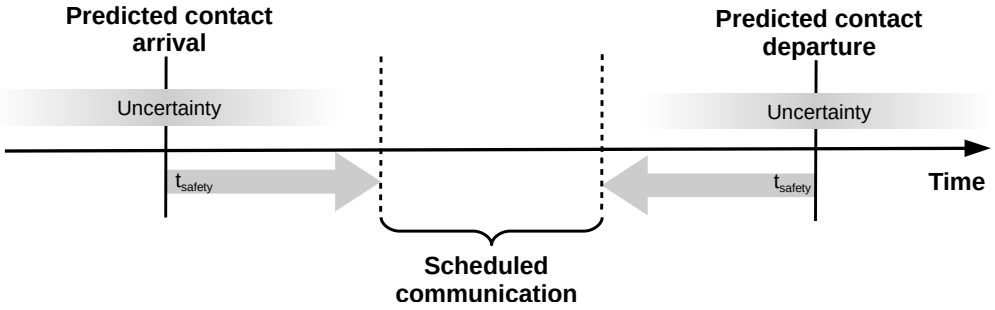


Figure 3.2.: Soft-scheduled contact with no discovery

might have a certain chance to be able to forward a packet to a destination sometime in the future based on past observations.

Figure 3.1 shows the relation between the different types of DTN contacts. No discovery mechanism is needed for *Persistent Contacts* or *Hard-Scheduled Contacts*: In the first case a node is assumed to be available all the time, and in the second case it is exactly known, when a node will become available, thus a sender can just defer transmitting data until a point in time where it knows the node will be ready for reception. For *Soft-Scheduled Contacts* the situation becomes more unclear: Imagine a bus-based DTN, where the timetable is known, but due to traffic conditions the actual schedule will differ slightly from the timetable. Without a discovery mechanism, one needs to include a safety margin, i.e. if according to the expected schedule communication is possible between times t_0 and t_1, a system needs will allow communication only between $t_0 + t_{safety}$ and $t_1 - t_{safety}$. Thus, the usable time of a contact would be decreased by up to $2 \cdot t_{safety}$ from the optimum as shown in Figure 3.2. Contact utilization is equal to a discovery mechanism that needs $2 \cdot t_{safety}$ time to detect another contact. However, no energy will be spent for discovery purposes. It is to be expected, that for most soft-scheduled contacts a discovery mechanism will lead to better utilization as t_{safety} needs to be chosen very conservatively to get a reliable system. For all contacts on the right side of soft-scheduled contacts in Figure 3.1, some form of discovery mechanism is required.

3.2.1. Contact Utilization

Contact utilization in a DTN is fundamentally limited by the speed of the discovery mechanism. A lower contact utilization can influence end-to-end throughput and latencies.

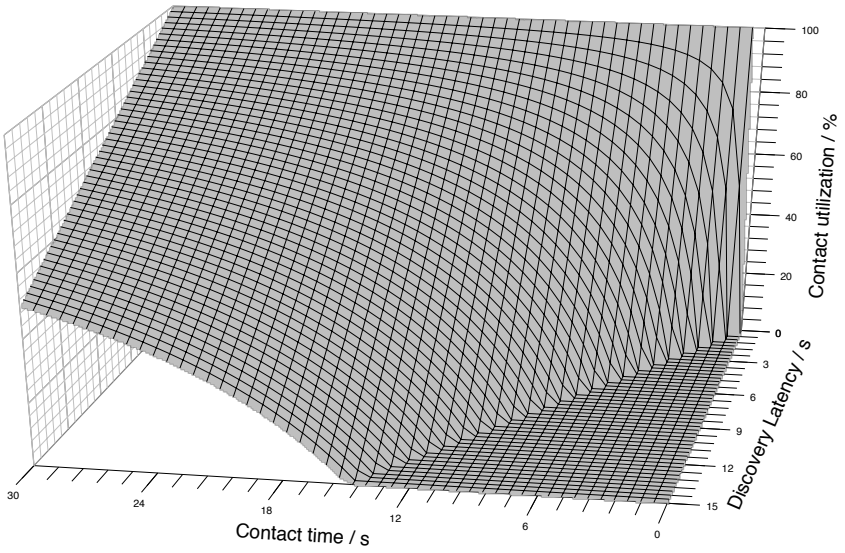


Figure 3.3.: Contact utilization vs. discovery latency

More specifically with

- bandwidth b in bytes/sec
- length of contact t in seconds
- transferred volume v in bytes
- discovery time d in seconds

when two nodes meet, the amount $v = b \cdot (t - d)$ (with $t > d$) of data can be moved. Divided by the maximum possible utilization that could be leveraged in the case of *Persistent Contacts* or *Hard-Scheduled Contacts* we get the contact utilization U

$$U = \frac{b \cdot (t - d)}{b \cdot t} = \frac{t - d}{t} \quad \textbf{with } t > d \tag{3.1}$$

The relation between contact utilization and discovery time is shown in Figure 3.3. Obviously, when $t \leq d$, a contact is useless and no data can be transferred. Therefore, especially for scenarios with short contacts due to movement or radio range, the discovery time d should be as short as possible. However, a common scenario with high mobility are Pocket Switched Networks, where battery-powered devices such as smartphones are used. In this case the goal of achieving shorter discovery latencies equals higher energy usage for the discovery process and thus a shorter battery life. Therefore, discovery latency and energy usage need to be balanced depending on the application.

3.2.2. Discovery for Wireless Mobile Devices

The basic principle of local discovery in wireless networks is always the same: Nodes broadcast presence information that can be received by other nodes. The frequency of

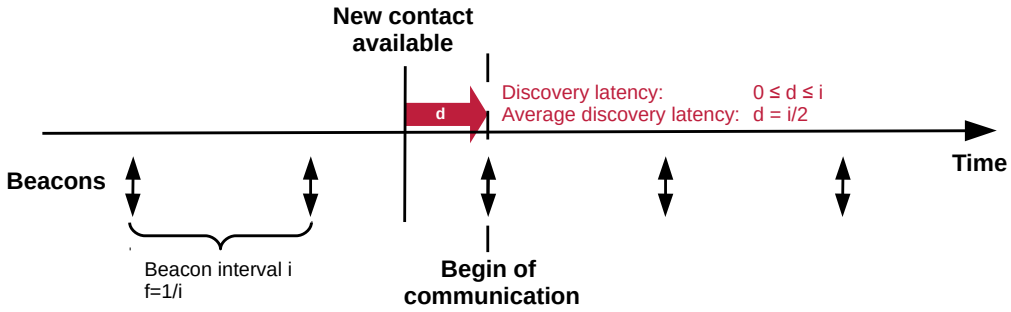


Figure 3.4.: Beacon frequency and discovery latency

beaconing is directly related to the discovery time d. If beacons are broadcast with a frequency f Hz, on average $d = \frac{1}{2 \cdot f}$ seconds can not be used for communication during each contact (see Figure 3.4). For example, sending beacons with 0.1 Hz will lead to an average of 5 s lost on each contact. While transmitting beacons with a higher frequency uses energy, a more severe problem is the listening power consumption. A node can always decide at which frequency and with which duty-cycle it will announce itself, however as a potential receiver a node must keep its RF hardware in a high power state at all times to be able to receive beacons. This is the reason why the IBR-DTN BP implementation on Android phones [16] used to disable or heavily restricts IPND by default to allow the Wi-Fi hardware to switch off when the phone is idle.

Potential solutions are simple duty-cycling methods, where the receivers know at which time to expect beacons. Now nodes can keep their RF hardware in a low power state and only go active when a discovery slot arises. In many practical scenarios this is unfeasible as it requires strict time synchronization between all nodes. Special schedules have been proposed to overcome this limitation and allow for asynchronous discovery even if nodes are duty-cycling their radios. In the *DISCO* [29] protocol nodes choose two prime numbers and enter a discovery phase consisting of either beaconing, listening or both in every time slot that is divisible by one of their primes. It is easy to see that the schedules of two nodes will overlap at some time, and taking the chosen primes into account an upper bound for the time required before two nodes synchronize can be derived. Related scheduling approaches with improved discovery latency over *DISCO* are *Searchlight* [30] or quorum-based schemes based on *Perfect Difference Sets* [31]. *Searchlight* has also been implemented and tested on real smartphones. These approaches promise combining discovery with duty-cycling for energy constrained nodes even when clocks are not synchronized perfectly.

In the common case of using Wi-Fi enabled devices such as smartphones a practical implementation hurdle is the long and device-dependent time to bring the Wi-Fi interface up and down [30]. This implies longer slot length, resulting in practically achievable energy efficiency that is less than optimal. While a higher duty cycle is preferable from a contact utilization point of view, the Wi-Fi implementations in current mobile devices are limited by the long transition times, and devices may keep the Wi-Fi hardware in a high power state for some time after the last transmission, which negatively affects energy efficiency.

As Wi-Fi uses a serious amount of power, many mobile phones or tablets switch off the Wi-Fi hardware when they are in standby and no application is actively transmitting data.

In light of these practical hurdles, it remains questionable, whether opportunistic applications relying on spontaneous interaction between battery-powered mobile devices are practically feasible in the foreseeable future. While aspects such a device's speed to cycle between power states and the general power-efficiency of a wireless technology can be improved in the future, the principal hurdles remain: Idle listening will always cost a significant amount of energy and the trade-off between discovery latency-success rate and energy usage will remain. Having an always-active discovery mechanism on a battery powered device will always have a significant impact on energy usage. Luckily, for DTN applications this is often not necessary.

Mobile DTN applications which do not require intermediate interaction can save a lot of energy when they opt to just use the Wi-Fi interface when it is active anyway due to the user using the device. While this would significantly reduce the chance of finding a direct neighbor through a discovery protocol, as both nodes must be actively used at the same time, data can still be exchanged reliably by using an always available DTN router as *Persistent Contact* in the Internet. The trade-off to be made here is between latency and availability. On the other hand, contact and thus energy utilization are getting better, as contacts are shifted from device-to-device contacts to persistent device-router contacts, which can always be utilized fully.

3.2.3. Discovery Using Secondary Radios

The principle that broadcasting and listening for beacons, especially when using high bandwidth radios, consumes a lot of energy can not be solved. However, using a low-bandwidth, low-power radio exclusively for discovery is a viable solution to save energy in some applications. One such system has been implemented by the Dieselnet project: A fleet of buses communicates via Wi-Fi with battery powered roadside units called throwboxes [32]. However, most of the time the Wi-Fi hardware and processing unit will be powered off. Only a small sensor node with a long-range, low-power 900 MHz ISM radio is active. The buses are equipped with a similar radio. When a bus comes in range the processing hardware and Wi-Fi interface of the throwbox will be powered up and ready once the bus comes into Wi-Fi communication range. A similar approach has also been used in other projects. For example, a flexible IEEE 802.15.4 based power management module for solar powered outdoor DTN nodes that can wake up the Wi-Fi powered hardware using a rule-set based on requests, priorities and current battery status was presented and experimentally evaluated in [33].

3.2.4. DTN Discovery in the Internet

While the beaconing-based discovery methods developed so far are only applicable for local networks, the question is what a suitable approach for Internet applications would be. Obviously, announcing an active node to "all" nodes in the Internet is not an option. One might opt for some sort of central registry that could be searched, however the

most sensible approach is just applying the pull-approach adopted by common Internet protocols: IP nodes do not announce themselves, instead connections are initiated by nodes that already know another node's IP address or Domain Name System (DNS) name. "Discovery" of other nodes as discussed above is only done on a much smaller scale in the application layer, i.e. when managing the roster of a messaging application or for game server browser. Similarly, for a BP-based DTN we should require that any BP node in the Internet can be contacted as long as its EID is known. Proactive discovery by means of beaconing is a useful tool for small networks, when energy is not an issue. For the Internet, or when energy is an issue, proactive discovery of all neighbors should be replaced with a demand driven ability to find any node or service when only its EID is known. Which EIDs are relevant or available to an application is out of scope for the BP layer, and is dealt with at the application layer. Just as in the Internet any available IP can be connected on demand, we want a mechanism that allows contacting any available BP EID without any special configuration or prior association with it.

3.2.5. Available Discovery and Name Resolution Approaches

For LANs or small-scale ad-hoc DTNs there exists a standard mechanism to detect neighboring nodes and services: IP Neighbor Discovery (IPND) [34]. IPND works by regularly broadcasting beacons, and thus, it is not suitable to be used in big multi-hop networks such as the Internet. In [35] Waldhorst sketches Arriba, a general architecture for routing in overlay networks spanning heterogeneous technologies based on generic node ids. Arriba focuses on routing, but it does not specify how to create unique node ids and assign them to device names or underlay addresses. Closely related to the problem of node discovery is the problem of routing: Most general routing protocols proposed for DTNs are designed for ad-hoc type scenarios and as such are often variants of flooding like Epidemic routing [36], MaxProp [37] or PRoPHET [28]. Other approaches exploit domain specific knowledge [38] and are thus not generally applicable.

Earlier BP-centric DTN specifications included the concept of "regions". DTN regions were a hierarchical naming concept for DTN nodes based on their network affiliation [39]. Current specifications have abandoned this concept in favor of a more flexible flat URI-based namespace. The idea is that different networks can be identified by different URI schemes, but generally the usage of URIs is meant to be much more open. The specification suggests things such as "expressions of interest" URIs [2]. To allow hosts to find out the network layer address for a host name in the Internet the DNS [40] is used. In addition to a number of shortcomings of traditional DNS in the Internet [41], it is also not optimal in a DTN. DNS is partitioned assuming a hierarchy of hostnames, which is not required by the flat URI-based namespace of DTNs. Furthermore, DNS is not self-organizing but instead it involves significant organizational overhead. It assumes a hierarchy of servers that are explicitly administrated, following the administrative hierarchy of the namespace. Also, DNS assumes that the network of servers is static and rather stable - a property, which cannot necessarily be found in DTN. To overcome some of these problems, DDNS [42] has been proposed. It is based on a Distributed Hashtable (DHT) as data storage and embeds

a hierarchical namespace in the flat key space of a hash table. DDNS tries to mimic the behavior of DNS as it is intended to be a DNS replacement, and thus it only transfers the DNS semantics, including hierarchy, to a DHT-based system.

3.3. Distributed Hashtables

As suggested in [43], DHTs might be a feasible way to tackle the naming problem in DTNs. DHTs are a robust way to store data in a distributed fashion. A DHT is a key-value store which is distributing the load evenly across participant nodes while still providing good lookup performance. Generally DHTs are resilient against node failure, have excellent scalability and support a flat name space.

The basic idea of a DHT is to construct an overlay network which maps areas of a flat hash-space to specific peers, while enabling routing to any peer using a logarithmic amount of steps in the overlay network. This general principle is shown in Figure 3.5. Every node is assigned a certain chunk of hash-space depending on its own ID. A node is responsible for storing key-value pairs corresponding to its area in hash-space. Nodes also know a certain amount of neighbors, enabling them to route queries outside their area of responsibility towards their destination. Different overlay structures such as rings or different types of trees or graphs are possible. The example in Figure 3.5 sketches a unidirectional ring-based DHT, where nodes know their successor nodes and a couple of shortcuts within the overlay ring. As such, when the node with the id 589 makes a query for a key 48678 the query is routed iteratively closer to the responsible node, which then returns the stored value.

DHT implementations differ in the topology in which they organize participant nodes, and in the metric used to determine which key belongs to which node. The most well-known DHT is Chord, introduced in the seminal paper by Stoica et al [44]. Chord uses a ring topology. In addition to successor information, each node has a routing table which enables $O(log(n))$ communication complexity while searching for a key. Among other well known DHT variants are Pastry [45], which tries to exploit local neighborhood information, and CAN [46], which organizes data in a d-dimensional grid. A more recent DHT is Kademlia [47] which is widely used in P2P networks such as BitTorrent (BT). A Kademlia DHT is organized as a tree using the XOR operation as distance metric. Due to Kademlia's properties, nodes automatically learn more about the DHT 's structures while routing and forwarding DHT messages. Thus, from all DHTs Kademlia has best proven the DHT 's alleged scalability and performance properties in real world scenarios [48, 49]. For a more thorough introduction to DHTs see [50].

3.4. NASDI: Naming and Service Discovery for Internet DTNs

As lined out in the beginning of this chapter, the established BP ecosystem has a major shortcoming when it comes to operating in large-scale fully interconnected networks such as the Internet: There is no standard mechanism to find a node or the next hop for a specific DTN EID. Compared to the standard IP architecture there is no standardized

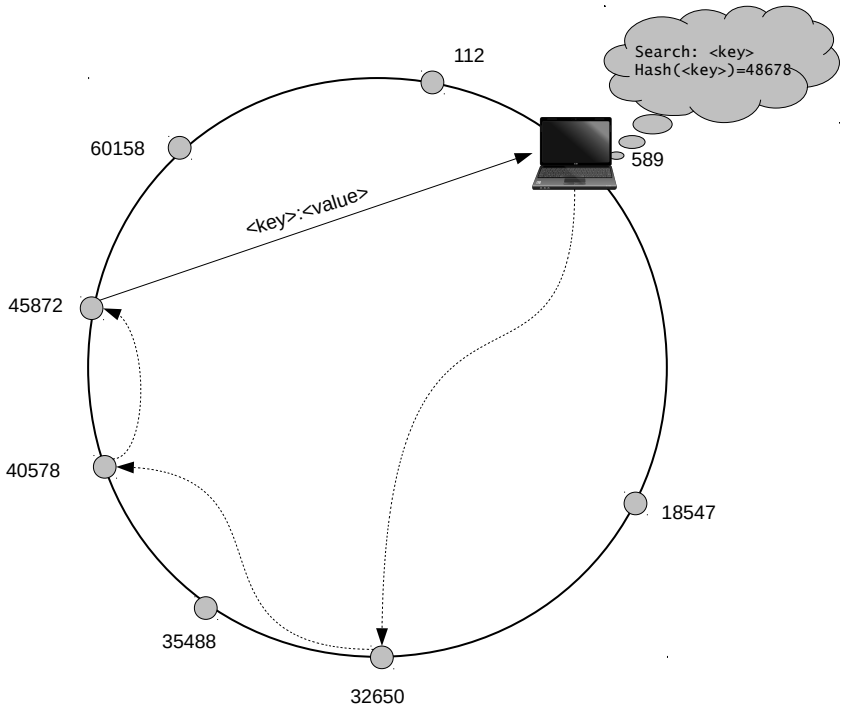


Figure 3.5.: Basic DHT example

naming system such as DNS and there is no usable routing protocol to find the next hop to a destination if that hop is located in a so far unknown network across the Internet. The various proposed routing protocols for DTNs normally assume an ad-hoc scenario relying on different forms of flooding and network discovery, both of which are not applicable in the Internet. In fact, the "DTNBone" [51], which is a collection of DTN nodes operated in the Internet by different institutions for DTN testing purposes, consists mainly of a webpage. This page contains (often inaccurate) information about which node can be reached at which IP address using which transport protocols. A DTN administrator who wants to connect with the DTNBone takes this information in order to configure a static route within his DTN server.

Therefore, we proposed and implemented Nasdi [23], a mechanism that allows for naming, service discovery and routing between DTN nodes operated in a large backbone network such as the Internet. Nasdi is able to integrate peripheral networks and nodes which are only intermittently connected to the backbone and it even allows nodes which do not implement Nasdi to take advantage of the benefits. As DTNs contain intermittently connected nodes that may enter or leave the network at any time, it is our goal to be notified about such events even if the node in question is not located in our direct network neighborhood. To facilitate this Nasdi offers an asynchronous notification mechanism. Nasdi is based on a DHT and is specifically adapted to the needs of DTNs.

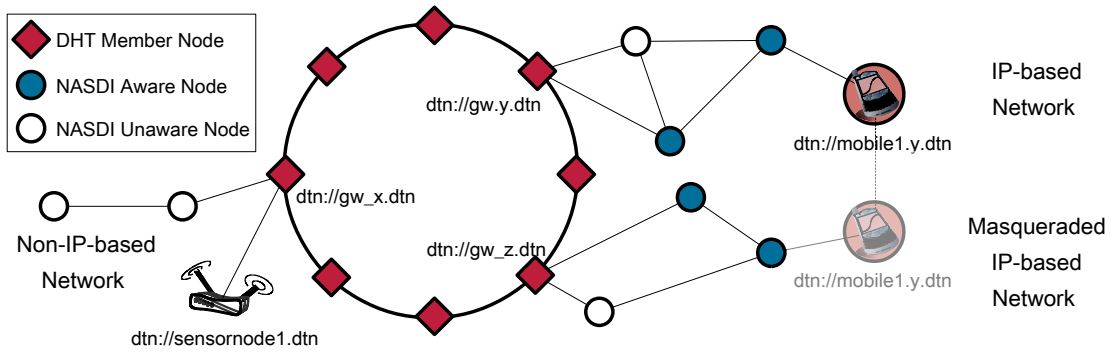


Figure 3.6.: Scenario overview

3.4.1. System architecture

Nasdi is a distributed system that can provide naming, routing and service discovery for Internet connected DTNs. Nasdi is based on a Kademlia DHT. Its goal is to help connecting to other DTN nodes based on their EID. As we will show, Nasdi does not necessarily find the destination itself, but possibly a suitable next hop that can be used to route to the destination. Due to the various, sometimes very application specific, routing protocols available for DTNs, unlike [35], Nasdi does not try to replace former routing protocols in DTNs. It rather augments them by bridging the gap between separate DTN networks connected through the Internet that cannot discover each offer by the conventional link local discovery mechanisms. For example, to reach a so far unknown node in another network, Nasdi provides the connectivity information of a suitable node. This node can then be contacted and bundles can be exchanged. Optionally, routing-packets for a mechanism such as PRoPHET might be exchanged with the newly discovered node to learn about the network behind it. However, the choice of a DTN routing protocol is independent from Nasdi.

An overview of a Nasdi System is depicted in Figure 3.6. Apart from providing naming services to well connected IP capable DTN nodes, the Nasdi architecture also allows to integrate peripheral networks that are not directly reachable or which use a non-IP transport layer for the BP. These networks can be transparently proxied by gateway routers as explained in Section 3.4.2. Additionally, Nasdi can also improve the connectivity of legacy nodes not supporting Nasdi.

3.4.2. DTN Node Roles

When Nasdi is deployed each node can assume different roles in the Nasdi system. The roles are depicted in Figure 3.6. It is to be expected that not all nodes will support Nasdi. Thus, when deploying Nasdi the following node types can be expected in the network:

- **Nasdi-aware nodes:** Nodes which implement Nasdi (filled circles and diamonds in Figure 3.6). These nodes can be full DHT members, i.e. storing information for the DHT or just query the DHT.

- **Nasdi-unaware nodes:** Nodes which do not know about the Nasdi mechanism (empty circles in Figure 3.6). Lots of unaware nodes are to be expected before Nasdi is widely adopted within the DTN community. Nasdi-unaware nodes can still be advertised within the DHT by a Nasdi-aware DHT member. They cannot, however, query the DHT for information. They can benefit from Nasdi by routing through a DHT member.

Nasdi-aware nodes can choose to become a **DHT member** (diamonds in Figure 3.6). A DHT member is responsible for storing data which is assigned to it by the DHT implementation. Additionally, it is responsible for regularly refreshing the DHT information of nodes it advertises in the DHT. Therefore, a node with high mobility or insufficient network connection might choose not to join the DHT. Such nodes will not be able to keep in constant contact with the DHT, leading to high network churn within the DHT which is detrimental to its performance. Contact information of nodes can be stored in the DHT in two different ways:

- **Announced nodes:** Announced nodes are nodes whose own convergence layer information is stored within the DHT. The convergence layer information stored in the DHT points to the node itself, i.e. contains its current IP address. A node can announce itself in the DHT if it is a DHT member, or it can ask a DHT member to announce it.

- **Proxied node:** Nodes in networks that are not accessible via IP from the Internet. They need a suitable DTN router in order to participate in the global network. Reasons for unreachable nodes might be a firewall or NAT router. Proxied nodes are nodes which have the convergence layer information of another DTN node stored in the DHT. This node might for example be an organization's central DTN Internet gateway. The proxy can also be used as a gateway between different underlying network technologies such as IP and ZigBee. Another rationale behind proxying nodes is that a node might only be intermittently available which leads to frequent DHT updates and inaccurate information. Instead, storing the convergence layer information of a node that is more likely to be online, allows other DTN members to route bundles in the correct direction while the proxy node is in a good position to relay the information to the target as soon as it is available.

Please note, that a node may be announced directly and at the same is being proxied by others. When querying the DHT all entries are returned, and a node can decide whether it will try to contact the requested node directly or whether to it chooses on of the advertised proxies.

3.4.3. DHT Information Management

This section details the information stored in the DHT and the steps needed to maintain and query the DHT. We assume that the DHT provides a method DHT_route(msg, key,

Key ABBC2134	
time_to_live	1000
time_since_last_seen	100
time_refresh_passive	200
type_entry	SINGLE
type_information_list	TCP, UDP

(a) val_{stored} single entry

Key 34FF4320	
time_to_live	200
time_since_last_seen	100
time_refresh_passive	110
type_entry	GROUP
type_information_list	AB21

(b) val_{stored} group entry

Key 0F43014C	
number_of_notifications	1
event	reoccur
interest	AB21

(c) $notify_{pend}$ entry

Figure 3.7.: Data stored in the DHT

value) which delivers a message of type *msg* with content *value* to the node(s) responsible for the partition of key space containing *key*.

The storage at DHT member nodes is a set val_{stored} of $(key, value)$ tuples. For a given key a number of values can be concatenated, which is needed for group management and is also a way to deal with duplicate names: As the BP forces no structure on the EID name space, it is valid and to be expected that for example multiple `dtn://test` nodes will join the network. Replacing is not an option because we do not want malicious or similarly named nodes to expunge valid entries from the DHT. Security critical applications which need to certify the identity of other nodes, can use the BP Security Protocol [52]. Therefore, spoofing other nodes in the DHT does not pose an additional risk.

Figure 3.7a shows an example entry for a stored node. The *key* is the key used as address in the DHT and it is derived from a node's id by SHA-1 hashing it. The *type_entry field* denotes whether this is a group or single node entry (the BP allows the same form of id to be used for either a group or a node). The *type_information_list* contains the IP address and port numbers of the TCP and/or UDP convergence layer for a single node entry, or a list of hashed node ids for a group entry.

Different timers are used to determine when to expunge an entry and to assess the freshness of the data:

- **time_to_live** (ttl): This timer determines how long this entry is considered to be valid. The initial value is determined by the node publishing the key into the DHT. The node that stores this entry decrements it. The ttl is a measure how long the contact information in this entry is still assumed to be valid.

- **time_since_last_seen** (tls): Even if the ttl is very high (e.g. for announcing a stationary node), the entry should be refreshed periodically. The tls counts how many seconds have past since the entry was last updated in the DHT.

- **time_refresh_passive** (trp): The trp value is constant and indicates how often the publishing node intends to refresh the entry. If $tls > trp$, this means that an entry was not refreshed within the expected time. This can indicate that the publishing node has connectivity problems. If a node announced itself, or if it announced itself to be a proxy for another node, there is a high probability that the destination may not be reachable at the moment even though the entry should still be valid according

to *ttl*. A Bundle Protocol implementation might opt to enable delivery reports in such a situation, especially if an unreliable convergence layer such as UDP is used.

3.4.4. DHT RPCs

The following DHT Remote Procedure Calls (RPCs) have to be implemented by DHT members. We assume that a *key* (which is used for DHT routing) and a *value* is associated with each message.

- **GET:** Standard DHT operation. Returns all entries for *key*. The *value* parameter is ignored for this call.
- **STORE:** Standard DHT store. Stores *value* associated with *key*. Existing entries for *key* are extended.
- **JOIN_GROUP:** Allows augmenting information stored for a *key* describing a group. Creates a new val_{stored} group entry if the group does not exist so far.
- **LEAVE_GROUP:** Deletes the node in *value* from the group entry designated by *key*. Does not touch other entries for *key*.
- **NOTIFY_REQUEST:** Indicates that the node id contained in *value* wants to be notified when a modification is done to *key*. A user should be able to specify whether this should be a "one-shot" notification, i.e. whether the notification request should be cleared after the notification is fired the first time or whether this should be a permanent notification request. See also Section 3.4.5.
- **NOTIFICATION:** This message contains a notification for the key *key*. If the current node's id is *key*, the notification is forwarded to the application layer. Otherwise, the $(key, value)$ tuple is stored.

The processing of these messages is shown in Algorithm 1.

3.4.5. Asynchronous Notification

In a mobility enabled DTN network nodes might not be reachable at all times. This is a standard case in DTN networks and participating nodes store bundles for an unreachable destination until a suitable next hop becomes available or the bundle expires. However, it is beneficial if the node storing the bundle is notified as soon as the destination becomes available again. This can be implemented using the DHT. To support notifications a DHT member node maintains a second set $notify_{pend}$. A $notify_{pend}$ entry is depicted in Figure 3.7c containing the following items:

- **key:** This is the DHT key of the node, we want to receive notifications about.
- **number_of_notifications:** How often this notification should fire. Typical values are 1 or ∞. For 1 the event fires once, and afterwards the $notify_{pend}$ entry will be deleted, for ∞ the event will fire every time its triggering conditions are met. Continuous notifications can be cancelled by storing the notification again with a value of 0.

Algorithm 1 Process messages

```
1:  procedure processMessage(msg, key, value)
2:      if msg = GET then
3:          return {(k, v) | (k, v) ∈ val_stored ∧ k = key}
4:      else if msg = STORE then
5:          entry ← (key, v) | (key, v) ∈ val_stored
6:          if entry == ∅ then
7:              entry ← {(key, value)}
8:          else
9:              merge_single(entry, value)              ▷ Add entry to existing key
10:         end if
11:         val_stored ← val_stored \ {(k, v) | k = key}
12:         val_stored ← val_stored ∪ entry
13:     else if msg = JOIN_GROUP then
14:         entry ← (key, v) | (key, v) ∈ val_stored
15:         if entry == ∅ then
16:             entry ← {(key, value)}
17:         else
18:             merge_group(entry, value)               ▷ Extend existing group
19:         end if
20:         val_stored ← val_stored \ {(k, v) | k = key}
21:         val_stored ← val_stored ∪ entry
22:     else if msg = LEAVE_GROUP then
23:         entry ← (key, v) | (key, v) ∈ val_stored
24:         if entry ≠ ∅ then
25:             entry ← remove_from_group(entry,value)
26:             val_stored ← val_stored \ {(k, v) | k = key}
27:             val_stored ← val_stored ∪ entry
28:         end if
29:     else if msg = NOTIFY_REQUEST then
30:         notify_pend ← notify_pend ∪ {(key, value)}
31:     else if msg = NOTIFICATION then
32:         if key = my_id then
33:             notify_app(value)
34:         else                                        ▷ Indirect notification
35:             notify_pend ← val_stored ∪ {(key, value)}
36:         end if
37:     end if
38:     if msg ≠ NOTIFY and msg ≠ NOTIFICATION then
39:         check_notify(key)                           ▷ See Algorithm 2
40:     end if
41: end procedure
```

Algorithm 2 Check and transmit pending notifications for EID

1: **procedure** CHECK_NOTIFY(EID)
2: $targets \leftarrow \{target \mid (EID, target) \in notify_{pend}\}$
3: $data \leftarrow \{(k, v) \mid (k, v) \in notify_{pend} \wedge k = EID\}$
4: **for all** *target* in *targets* **do**
5: DHT _Route(NOTIFICATION, *target*, $(EID, data)$)
6: **end for**
7: **end procedure**

- **event:** Defines, which kind of event triggers this notification. Possible triggers are the reappearance of a node, the change of any value in the *val_stored* entry for *key*, or the change of a specific value.
- **interest**: The key of the node that wants to receive a notification when this events fires.

To demonstrate the different steps of establishing a notification request and the further processing, we will look at an example from Figure 3.6: Assume that in Figure 3.6 the node *dtn://mobile1.y.dtn* is proxied by *dtn://gw.y.dtn*. When *mobile1.y.dtn* becomes unavailable, this will be detected by *gw.y.dtn* and the corresponding entry will expire in the DHT. However, foreign nodes might still be sending bundles for *mobile1.y.dtn* to *gw.y.dtn* because they used a cached older entry with a higher *ttl*, or the bundles have been sent before the DHT entry expired. Thus *gw.y.dtn* stores a NOTIFICATION_REQUEST for *mobile1.y.dtn* into the DHT issuing the DHT command

```
DHT_ROUTE(NOTIFY_REQUEST, dtn://mobile1.y.dtn, dtn://gw.y.dtn)
```

This request will be routed to the same node that is responsible for the key *dtn://mobile1.y.dtn* in the DHT key-space. Whenever a DHT member receives a call that creates or modifies tuples in its val_{stored} it will check whether there are any notification requests pending for the modified key (see Algorithm 2). In our example *mobile1.y.dtn* joins another network and gets itself proxied by *dtn://gw_z.dtn*. This means *dtn://gw_z.dtn* will issue a STORE to the DHT:

```
DHT_ROUTE(STORE, dtn://mobile1.y.dtn, conv_layer(dtn://gw_z.dtn))
```

The node responsible for the key *dtn://mobile1.y.dtn* will check its pending notifications for this key and finds *gw.y.dtn*. Instead of directly trying to contact *gw.y.dtn* it will use the DHT :

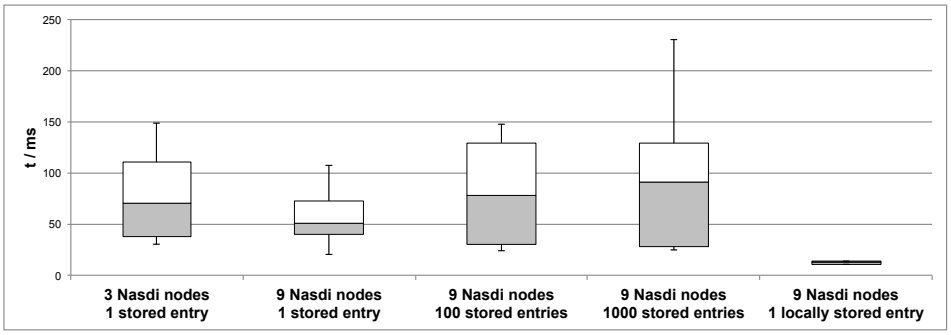


Figure 3.8.: Lookup times

```
DHT_ROUTE(NOTIFICATION, dtn://gw.y.dtn, conv_layer(dtn://gw_z.dtn))
```

If *gw.y.dtn* is online, this has the same effect as contacting *gw.y.dtn* directly, because we assumed that we use a DHT where each node is responsible for a range in keyspace containing its own node id. If *gw.y.dtn* is currently not available, the notification is stored on another node currently responsible for the key *dtn://gw.y.dtn*. Once *gw.y.dtn* becomes available again and rejoins the DHT, depending on the DHT, the mechanism of the underlying DHT will hand over the data for its chunk of the key-space, including the notification. This ensures, that receivers are notified as early as possible.

3.4.6. Implementation and Evaluation

While Nasdi is not a routing protocol in a strict sense, it could be implemented as such using the routing module interface of DTN2 [2]. For IBR-DTN [15] a new discovery module was the best choice for integration. IBR-DTN allows different submodules to plug into its event-based core. DTN2 offers an XML-based interface for external routing implementations. We implemented Nasdi for IBR-DTN using the Maidsafe library[3] which provides a Kademlia implementation including NAT traversal capabilities. The Nasdi implementation is an external program using Maidsafe which communicates locally via TCP/IP with a new IBR-DTN discovery module that acts as a wrapper for the external Nasdi implementation. This setup allows for great flexibility while developing Nasdi and should make it relatively easy to connect Nasdi to DTN2's external interfaces.

DHT Functionality Tests

While the performance of large scale Kademlia deployments has already been examined, e.g. in [49], we performed some small scale tests, to verify that the Nasdi implementation is working as expected. We used several virtual machines on the same local network running instances of Nasdi and IBR-DTN. Therefore, the results are not influenced by the performance or reliability of the network between the machines. The first Nasdi instance

[2] http://sourceforge.net/projects/dtn/

[3] http:/code.google.com/p/maidsafe-dht/ now merged into https://github.com/maidsafe/MaidSafe-Routing

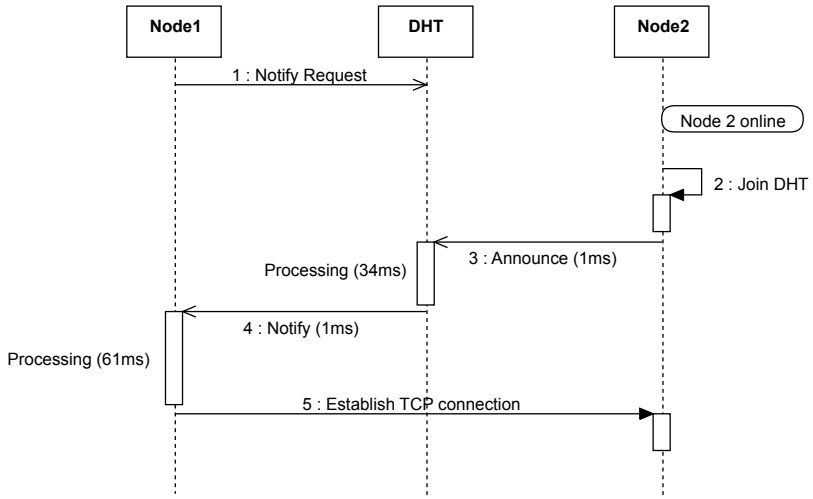


Figure 3.9.: Notification latencies

is always started standalone while the following instances get one of the running instances as bootstrap partner.

Lookup Test

Figure 3.8 shows the time to lookup an existing key in the DHT. The diagram includes the min, max and median values as well as the $Q_{0.25}$ and $Q_{0.75}$ quantiles. We varied the number of Nasdi nodes and the amount of tuples stored in the DHT. When the number of stored elements is increased from 1 to 1000 the average response time goes up from $\sim$ 70 ms to $\sim$ 90 ms, which shows the additional processing overhead in the Nasdi instances. The variance for each measurement is due to the fact that the DHT structure is different between runs, so that the responding Maidsafe instance might be nearer or further away. The rightmost plot shows the situation, when a node can answer the query from its local storage without the need to contact other DHT members.

Notification Delay Test

For this test we used 9 Nasdi instances. The IBR-DTN node *Node-1* was started sending a bundle to *Node-2*, which was currently not available. This leads Nasdi to store a notification request. Subsequently, we started IBR-DTN node *Node-2*. The Nasdi instance for *Node-2* announces its contact information in the DHT, which leads to a notification being dispatched to *Node-1*, which in turn connects to *Node-2*, delivering the stored bundle. We measured the time between starting of *Node-2* and the instantiation of a TCP connection to *Node-2*. A breakdown of the used time can be seen in Figure 3.9.

As can be seen in this case the notification itself is nearly instant, while the largest amount of time is spent after the notification in the IBR-DTN daemon getting the cached bundle from storage and preparing it for transmission.

3.5. DTN-DHT: Practical Naming for Internet DTNs

While the Nasdi approach presented in the previous sections represents a powerful and flexible mechanism to tackle the naming and routing problem for Internet-wide DTNs, implementing it poses some practical challenges: While the chosen DHT library is very powerful, it is also quite complex and introduces many additional software dependencies. Also, Maidsafe is not a standard library shipped with any operating system. Integrating the whole implementation into the main IBR-DTN branch would put disproportionately too much code into the codebase that needs to be maintained. Also, porting to our embedded targets, such as routers running OpenWRT, would be hard or impossible. Even when replacing the DHT library with a smaller, more lightweight option, the problem is still reaching a critical mass: The BP Community is not yet very large, only part of it deals with IP-based DTNs. Therefore, a DHT system such as Nasdi would initially only consist of very few nodes, which is not enough to ensure a stable and reliable DHT. The complete, publicly known "dtnbone", is only tens of nodes. Not all are reachable and not all of them run IBR-DTN. Often tested nodes might not be connected to the Internet directly. A robust DHT on the other hands needs an abundance of nodes to perform well. From experience with classical filesharing networks, where DHTs first found widespread application, at least hundreds or even thousands of nodes are desirable.

Therefore, Nasdi was completely redesigned to provide a more practical solution to the problem. The new approach is lightweight enough to be integrated into IBR-DTN by default and can work reliably as soon as two nodes decide to use it. We still wanted to keep a DHT as basic data structure for its flexibility. Instead of deploying our own DHT we decided to piggy-back our data on the DHT of an existing application to solve the problem of having enough peer nodes. We choose to adopt the DHT used in the widespread BT protocol. Building on the experiences from Nasdi we designed and implemented a naming system for BP-based DTNs that uses the BT DHT to provide the mapping from DTN EIDs to convergence layer information. Using an existing DHT has various advantages: Many stable and proven implementations are available, and there are always thousands of usable DHT members online, making the system more reliable even when the DTN Naming service is only deployed on a few nodes. The implementation presented here was eventually integrated into the mainline IBR-DTN source and the distributed prebuilt binary packages[4].

3.5.1. Architecture Overview

In BT the DHT is used to look up sources for chunks of a downloaded file. However, such a key lookup provides less usable information than what was specified in Nasdi. In fact, the only response the BT-DHT offers when querying for a key is a set of IP addresses and port tuples. Therefore, in contrast to Nasdi we developed a two-phase approach: EIDs are used as keys for the DHT and the IP of the node claiming that EID (or a node proxying that DTN node) are put into the DHT. An important detail in the BT-DHT is, that while it

[4] DTN-DHT was first shipped with stable version 0.8 of IBR-DTN released June 1st 2012

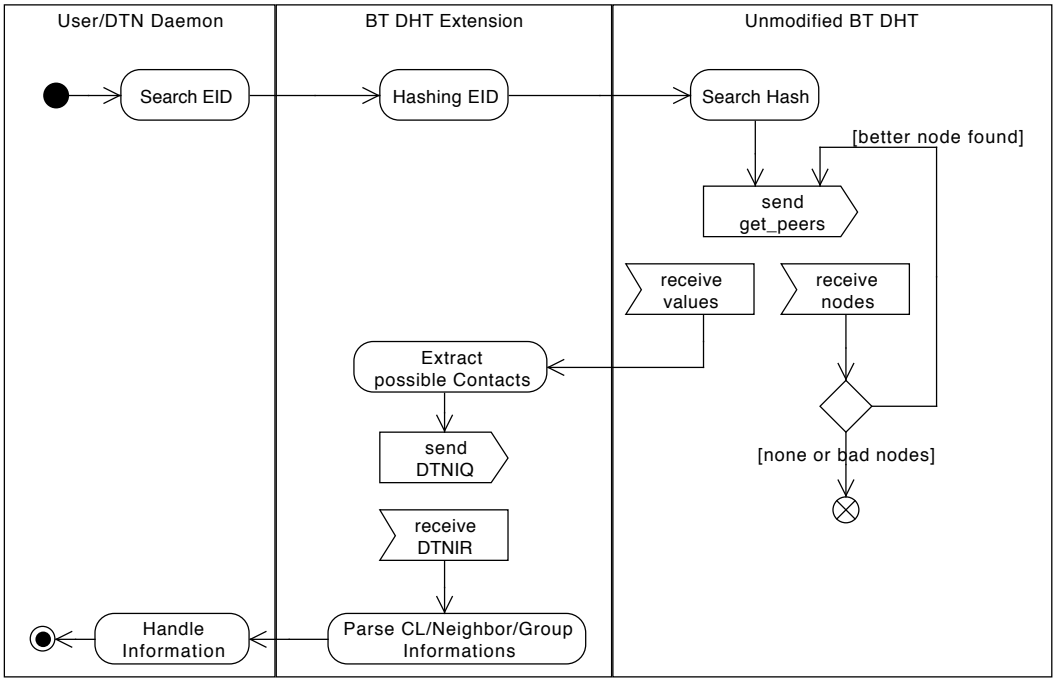


Figure 3.10.: BT-DHT components

stores IP addresses, it is not possible to actively store an arbitrary address. The protocol is designed in such a way that always the IP of the node issuing the store RPC is saved. Name collisions on the other hand are not a problem, as the BT protocol deals with multiple sources for one key: It is not possible to overwrite or evict an IP from the DHT by storing the same name. A store RPC for a given key only adds a value to the set of that key.

We developed a two-phase approach: Once an IP for an EID is found using the unmodified BT-DHT protocol (phase one), a second handshake is initiated, getting information relevant to the BP (phase two). The general architecture and operation of our approach is depicted in Figure 3.10. The system is composed of three parts: We use a fully compatible standard compliant BT-DHT implementation as basis. On top of this standard DHT we implement some extra functionalities that realize the DTN naming service. These extensions do not compromise compatibility with existing BT-DHT implementations. This extended DHT service can then be used by applications such as BP daemons to resolve and announce BP EIDs.

3.5.2. The BitTorrent DHT

The standard BT-DHT protocol [53] is a Kademlia[47] DHT. As the BT Kademlia variants are widely deployed, their performance has been studied in detail and their properties are widely understood [54].

The BT protocol uses SHA-1 [55] hashes to identify files, which will be shared by a user's BT application on a specific port. A node can store his local BT port into the BT-DHT

using the SHA-1 hash of a shared file as key. Nodes storing the port information also save the IP address from the node where the UDP store message originated from. For each key it manages, a node stores tuples of originator IP and the port stored by that IP. On a lookup all, or a subset, of the stored IP and port combinations should be returned. There is no method to delete an entry from the DHT. Instead, the BT-DHT employs a soft-state approach: All entries should time out after 30 minutes. Therefore, the store method has to be executed regularly to keep an entry alive in the DHT.

The BT-DHT uses a lightweight RPC protocol. The RPC messages in the BT-DHT are "bencoded" [56] ASCII messages sent through UDP. This encoding is also used by the normal BT protocol. Basically it is a length-prefixed binary encoding capable of storing strings, integers, lists and dictionaries. There are several open source BT clients with DHT support available. A widespread, platform independent open source client is Transmission[5] which we have chosen as the basis for this implementation. It utilizes a very lightweight BT-DHT implementation written in ANSI C[6].

3.5.3. BitTorrent DHT Protocol Extension

For a BP naming service there are several kinds of information that a user might want to store and retrieve: The most important information is the convergence layer address for a given EID. For the mostly used TCP and UDP convergence layers this includes the IP address and the used port. A node might provide information about more than one convergence layer. Additionally, information about a node's neighbors and joined groups are of interest. However, the BT-DHT does not allow much flexibility when storing data: The only data a user can reliably store in the BT-DHT is a port number (for BT this is the port, the BT daemon is listening on). Even an arbitrary IP address can not be stored directly in the BT-DHT. Instead, the IP is automatically taken from sender of a DHT store RPC.

Changing the normal BT-DHT routing operations or augmenting the store RPCs was not an option, as we want to remain compatible with the BT-DHT to be able to leverage the services of all BT-DHT members. Therefore, we choose to extend the BT-DHT protocol with new RPC calls in a way that retains compatibility with existing BT clients. We call the extended BT-DHT supporting the additional RPCs, needed for looking up EIDs, DTN-DHT. When storing information to the DHT, DTN-DHT nodes will use the EID of the node or group they want to store information about as hash and store the listening port of their DHT implementation (instead of the BT daemon port normal BT clients store). This structure is depicted in Figure 3.10: The rightmost part represents an unmodified BT-DHT implementation: It uses the standard *get_peers* RPC of the BT-DHT, which either returns nodes running a DHT implementation, which are nearer to the queried key in the DHT topology, or it returns the actual (IP, port)-tuples that have been stored for the queried hash. The middle part of Figure 3.10 contains the two added RPCs and the left side represents the application using our DTN-DHT implementation, usually a DTN daemon.

[5] `http://www.transmissionbt.com/`

[6] `http://www.pps.univ-paris-diderot.fr/~jch/software/bittorrent/`

Upon querying the DHT for an existing and announced EID, the DTN-DHT implementation should receive the IP and port information of one or more DTN-DHT implementations that stored this entry (the received *values* returned by the DHT implementation in Figure 3.10). Until this point, our naming service behaves exactly like a vanilla BT-DHT client. However, in the next step the querying node sends a newly defined DHT RPC, a DTN Information Query (*DTNIQ*), to all the nodes from the query's result set. In accordance with the bencoding defined in [53] a *DTNIQ* looks like this

bencoded:
```
"t":"aa", "y":"q", "q":"dtn",
"a":{"eid" :  "dtn://my_hostname"}
```

That query consists of the following elements:

- **t:aa**: A 2 byte transaction ID used to map queries to answers
- **y:q**: Denotes that this message is a *query*
- **q:dtn**: Identifies this query as the newly defined type *dtn*
- **a**: Array of arguments in the format *name:value*. So far only the source EID of the querying node is sent

If this query is sent to a node, which in fact is not a DTN-DHT node (because there is an invalid entry in the result set), that node will just ignore the UDP RPC query. Since most BT nodes advertise a TCP port for their BT service, a UDP message will never reach the other user's BT application. Should the other party run a UDP variant of the BT protocol on the advertised port, it will abort parsing the unknown message and silently drop it. We did not observe any problems when sending this unknown message to a normal BT client. As the *DTNIQ* is a syntactically correctly bencoded BT-DHT RPC, it allows parsers to abort parsing of this message gracefully. In fact, since there is a lot of software of varying quality or even downright malicious attackers [57, 58] to be found in the BT cloud, most implementations are pretty hardened to deal with all kinds of broken, garbled or unexpected messages.

If the query is received by another DTN-DHT node, it will answer with a DTN Information Response (*DTNIR*) containing information about itself. A *DTNIR* looks as follows:

bencoded:
```
"t":"aa", "y":"r",
"r":{"eid":"dtn://my_hostname" ,
"cl":["name=TCP;port=4556","name=UDP;port=4556"],
"nb":["eid1","eid2", ...  ],
"gr":["eid1", "eid2", ...]}
```

The response contains the following elements

- **t:aa**: A 2 byte transaction ID used to map queries to answers.
- **y:r**: Denotes that this message is a *response*.
- **r**: The EID of the answering node. This is important because it might be different from the key searched in the DHT, if the node was also announced (proxied) by its neighbors. For each key there might be one entry of the queried node and one or more entries announced by neighbors of that node.
- **cl**: A list of supported convergence layers. So far only formats for TCP, UDP and Mail (see Section 3.6) have been defined, however in the future it is easy to extend the list of supported convergence layers.
- **nb**: A node can opt to also disclose its neighbors. This way, the querying node learns more about the network (see Section 3.5.4).
- **gr**: A node will lists the groups it joined, if any. Doing so allows group-aware routing protocols to act accordingly.

The *DTNIQ*/*DTNIR* handshake also makes sure that only connectivity information of valid DTN nodes are given to the DTN daemon. This is important because we noticed that answers to DHT searches frequently contain, among the correct data, also bogus values. We analyze this in further detail in Section 3.5.6. As those non-DTN nodes will not answer the *DTNIQ* handshake, they will not be given to the DTN daemon, preventing it from making futile connection attempts to invalid nodes.

3.5.4. Routing and Neighbor announcement

To enable better connectivity in a DTN, we allow a node to not only announce its own EIDs to the DTN-DHT, but also publish EIDs of neighboring nodes. There are several reasons why this might be desired. A node could use a DTN implementation that is not DHT-aware. If such a legacy node is discovered by a DHT-aware neighbor using a mechanism such as IPND [34], it will improve connectivity of the network if that node announces its neighbor in the DHT, because it will make the legacy node available in the whole Internet by using the DHT-aware node as DTN router. Another common case is a node behind a NAT or firewall, which precludes it from being reached directly or announcing itself in the DHT. In this case its EID might be announced by another node in the same network which is reachable from the Internet. For example, an Internet router can announce DTN nodes within the LAN.

As the Bundle Protocol is transport agnostic and can span heterogeneous networks, a DTN node might be part of a non-IP network such as a IEEE 802.15.4-based wireless sensor network running μDTN [13] or an AX.25-based DTN2 network. In this case an IP-capable gateway can announce the non-IP based EIDs in the DTN-DHT, which allows for technologically heterogeneous DTN networks to be connected through the Internet.

If DHT nodes announce each other as neighbors reciprocally, a participant can get a good idea of the network structure by recursively querying the DHT for all nodes that

some node announces as neighbor. Sometimes it may not be desired that a single DHT node exposes a whole network by announcing all known neighbors. Therefore, apart from disabling the neighbor announcement completely, in IBR-DTN we support a new IPND service field which allows a node to opt-out from the neighbor announcement process.

It can be discussed whether the naming approach described here augments or replaces common DTN routing protocols in the Internet. The version shipped with IBR-DTN can be used in both ways: Without using other routing protocols, the implementation can be configured to add static routes to not-yet known neighbors of found nodes on the fly, e.g. once a node has been returned by the DHT and its information queried, the node and its neighbors are immediately reachable. In this case, a network of nodes connected directly to the Internet does not need any further DTN routing protocol. However, depending on the application it might be useful to combine the DHT mechanism with one of the existing DTN routing protocols such as PRoPHET. In this case a node returned by the DHT will be announced as new DTN neighbor using IBR-DTN's event system. This will trigger the routing module to exchange routing messages with the newly found node. This is useful if there is an intermittently connected multi-hop DTN network that is attached to the Internet by means of a central gateway running the BT-DHT service.

3.5.5. Security Considerations

As explained in Section 3.5.2, in case of conflicting entries the DHT will store all entries. This makes spoofing a node very easy. However, as the BP does not specify any structure for the EIDs and since we do not want to introduce an unnecessary central point to the system, the uniqueness and authenticity of entries can not be guaranteed. In fact, according to the BP it is quite valid (and to be expected) that many nodes with the EID *dtn://test.dtn* exist. This is not a problem. For applications where tighter security is needed the BP Security Protocol [52] can be used to verify the authenticity of the communication partner when establishing the link. In this case, spoofing nodes can only lead to longer delays while a node is checking all prospective communication partners. This might be abused for a DoS attack, by injecting many conflicting entries into the DHT, however in an open public system there are always suitable vectors for a DoS attack. Also, using the DHT does not preclude the daemon's administrator from configuring static routes to important communication partners.

3.5.6. Evaluation

We looked at the performance and reliability of storing information in the DTN-DHT. For the evaluation we always assume the worst case scenarios: Each test has been done with a new random DHT id and a new UDP port to prevent contacting to or being contacted by previously known DHT nodes. Also, unless otherwise noted, the boot strapping method is a basic DNS request, which means for each run it will take some time populating the routing tables.

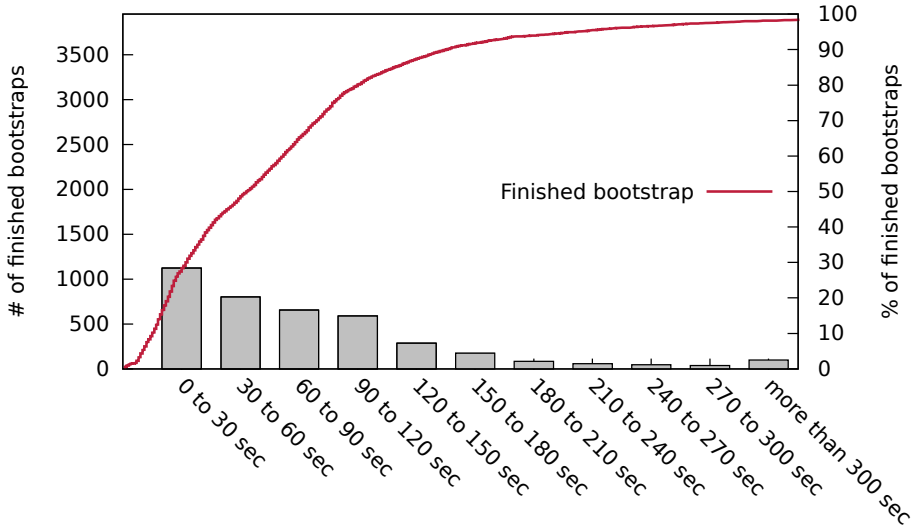


Figure 3.11.: Bootstrap times cumulative distribution function

Boot Strapping

We measured the DNS bootstrapping method to show how long the process takes without any prior knowledge of the existing DHT. The duration of the bootstrapping process is the time between initializing the library and knowing at least 8 DHT nodes. After performing 3954 test runs (see Figure 3.11), we see that in more than 95% of the cases, the bootstrap process is finished after at most 4 minutes. For nearly 50% of the test cases, the bootstrap process has finished after 60 seconds. The median duration was 63 seconds and the maximum observed duration was 1797 seconds.

These results show that generally boot strapping is no problem, however joining the DHT might not be a good option for devices with high mobility or intermittent network connection, such as mobile phones. When mobile phones change networks or switch off their Wi-Fi while not in use, they might not be able to join the DHT fast enough, or lead to high network churn within the DHT which hampers system performance. For such devices it is desirable to be announced by neighbors, such as stationary routers, which have a more persistent Internet connection. Then these devices can also act as DTN routers and buffer bundles for the destination while it is offline.

Lookup Success

We checked how reliably we can retrieve announced entries from the DTN-DHT. To make the different tests comparable, for each lookup the following sequence was performed:

1. Generate a random EID
2. Start a new DHT member node (using a new DHT-id and port)
3. After the bootstrap is finished, announce the EID

# Correct	Occurrences	Ratio
8	3	0.6%
7	29	5.8%
6	45	9%
5	69	13.8%
4	94	18.8%
3	102	20.4%
2	93	18.6%
1	42	8.4%
0	23	4.6%

Table 3.1.: Number of correct answers in 500 lookups

4. After the announcement finished, shut down the DTN-DHT node
5. Start a new DTN-DHT node (using a different DHT-id and UDP port)
6. After the bootstrap finishes, start a lookup for the EID generated in step 1

The DTN-DHT nodes are bootstrapped with a DNS query and with differing ids in steps 2 and 5. This test setup prevents DHT nodes to benefit from a good neighborhood from earlier runs. The announcement will store the SHA-1 of the given EID on 8 DHT nodes. Ideally, the same 8 DHT nodes should be found by the new DHT node making the lookup. But in reality, most of the time not all replicas are located.

We executed the test sequence lined out above 500 times and listed the occurrence of correct answers on Table 3.1. Only in 0.6% of the runs all 8 replicas are returned. In more than 95% of the runs, at least one entry is found. In nearly 5% of the lookups, the announced EID has not been found. However, in a real deployment this result is not very problematic: A DHT node that is announcing an EID will periodically refresh the entries. This will for example generate new replicas at other nodes if some nodes already left the network. Also, a longer running node is better connected in the DHT and therefore has a higher chance to find better neighbors to store the data. The same applies to the querying node: As long as it has some bundle for a EID for which it has not yet located an appropriate DTN neighbor, it will repeat the lookup.

So in real deployments we observed that for a small fraction of queries it takes a few minutes longer to find a published entry in the DHT, but eventually the correct entry will be found. Figures 3.12a and 3.12b give some insight into the general duration of lookups. The graphs show, how long it takes until a query for a previously published key yields the first, second and third correct answer. Usually most queries can be answered in less than a minute, but in some situations it is harder to find an entry, and especially finding a second or third correct answer can take a considerably longer time. The Figures also show the number of invalid results returned. Invalid results are entries which we get for the queried

key from the DHT which are different from the values we published. We take a closer look at the wrong results on page 40.

Entry Lifetime

According to the specification [53] all information published in the DHT should be deleted after 30 minutes. This prevents nodes from providing outdated information. Whether this actually happens depends on the individual DHT implementations of the participating nodes.

To measure the lifetime of an entry, a random EID was stored in the DHT as explained in Section 3.5.6. Continuous lookups have been performed until the entry vanished from the DHT. For most of the entries we found a lifetime of 30 minutes as can be seen in the example from Figure 3.12a. In some of our experiments we found, that there seem to be some DHT implementations out there which are saving entries for a longer period. This can be seen in the test results depicted in Figure 3.12b, where even 47 minutes after the last announcement the entry is returned.

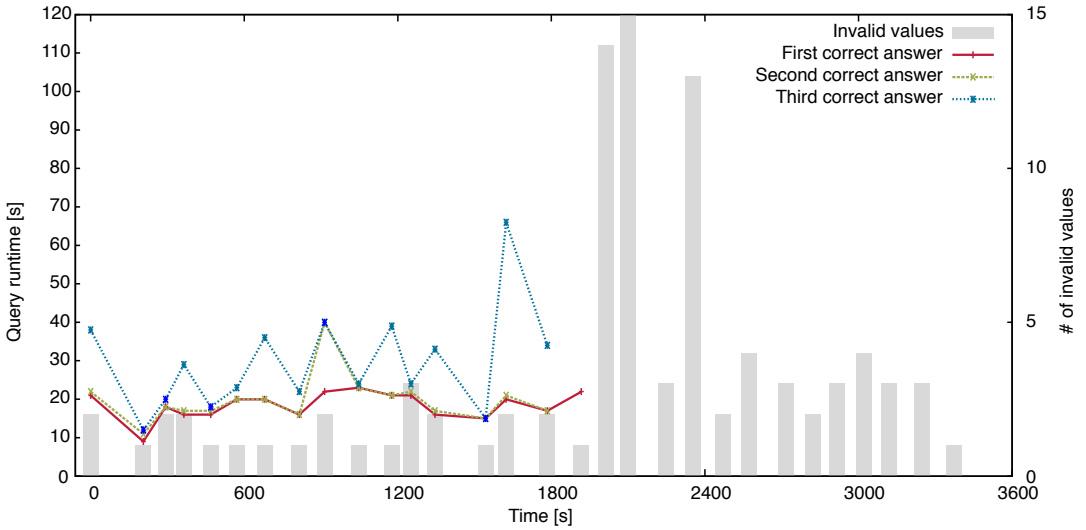
To get a better understanding of the average lifetime of entries in the BT-DHT we announced 500 EIDs and performed regular lookups every 5 minutes. The results of these tests can be seen in Figure 3.13. The success ratio shows how many lookups have been successful. We define successful as returning the correct entry at least once. The candle sticks show the median time until the first result for a successful query arrives, the interval in which 90% of the successful queries get their first results as well as the minimum and maximum duration until receiving the first result. It can be seen, that in accordance with the BT-DHT specification, after about 30 minutes, the success ratio is in sharp decline and falls to about 25%. This demonstrates that the majority of DHT nodes honor the requirement to drop data 30 minutes after the last refresh, but there are also a lot of implementations out there which will store values for a longer time.

Announcement

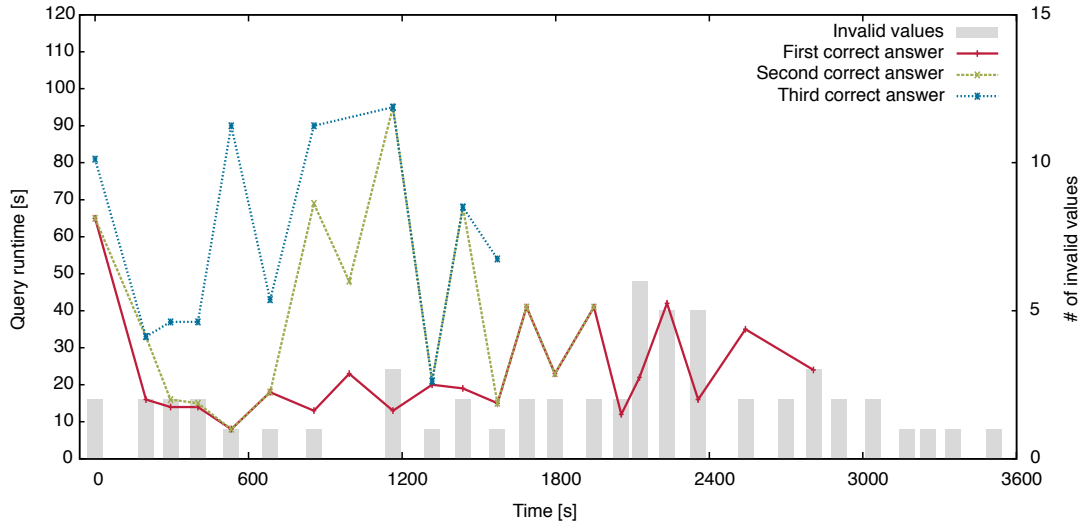
Before a freshly started DTN node can be found by other nodes, its information has to be announced in the DHT. An announcement is done in two phases: First there is a lookup for the key that should be stored (see Section 3.5.6) to get into contact with the DHT nodes which are responsible for storing that key. After the lookup phase the 8 nodes which are nearest to the key will be contacted by an RPC and asked to store the key.

We evaluated how long it takes to announce a key. We always measured the first announcement of a given key, which means the time to lookup the 8 responsible nodes will be high. Upon subsequent refreshes, the lookup phase will be much shorter, because the relevant nodes are already in the neighbor tables. This is consistent with the results from the lookup evaluation in Section 3.5.6.

We measured announcement times for a duration of 8 hours on 2 different days. We found the announcement time to be independent of the day or time of day. Therefore, Table 3.2 shows the distribution of announcement times from both days. The table shows that the first announcement of an EID after starting a BT-DHT node takes about 5 minutes,



(a) Fast lookup, correct lifetime



(b) Slower lookup, longer than expected lifetime

Figure 3.12.: Lookup delay and lifetime

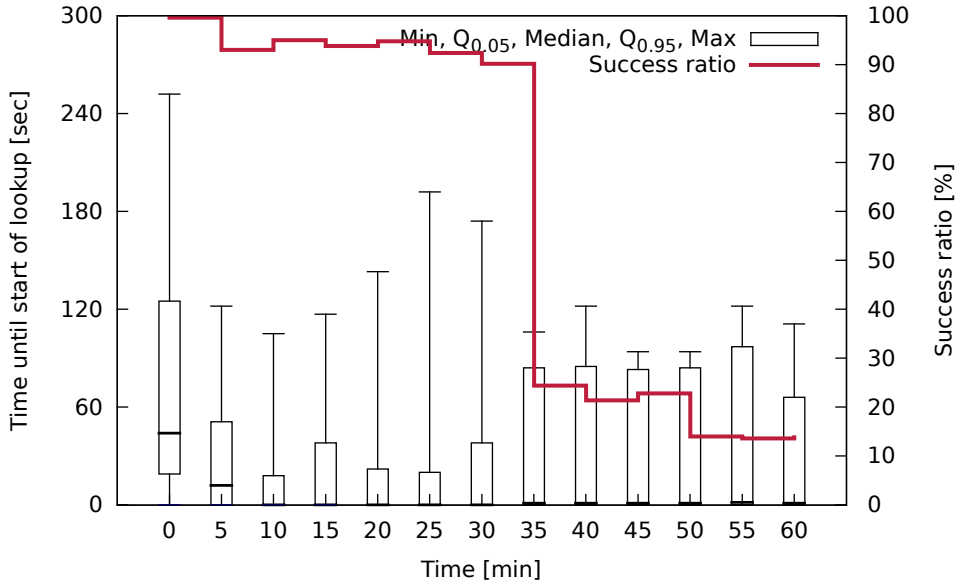


Figure 3.13.: Average entry lifetimes for 500 random EIDs

with a variance of two minutes. This is the duration of the lookup plus the time needed for 8 successful RPC calls.

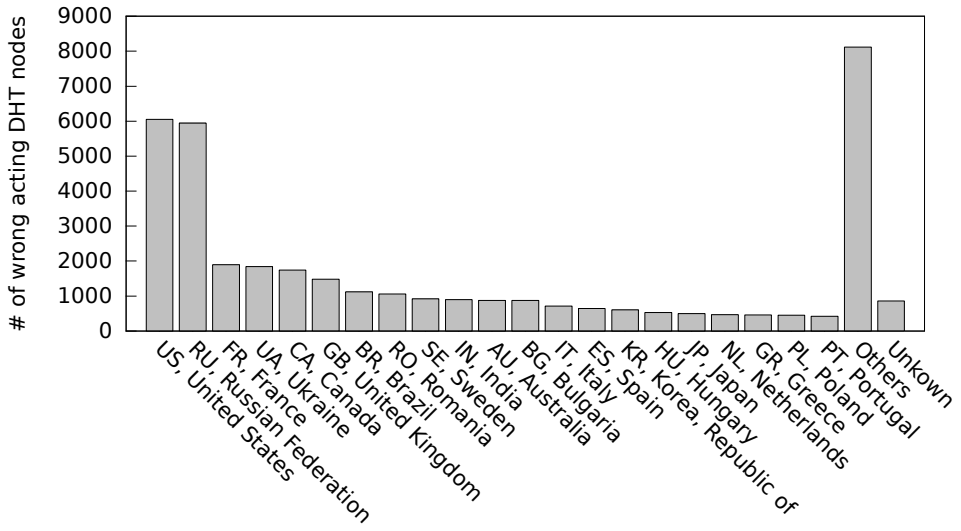
Invalid Contact Informations

Implementing and testing the BT-DHT naming service showed that the DHT contains a lot of nodes, which are not acting like they should. We observed many nodes, which are sending wrong answers on lookups. We define a wrong answer as receiving IP addresses for an announced hash that are different from the one we published. Due to SHA-1's big key space it is highly unlikely that this is caused by a random hash collision. The amount of bogus answers can be seen as gray bars in Figure 3.12a and 3.12b. It seems, many nodes answer with invalid contact information even when querying non-existing SHA-1 keys.

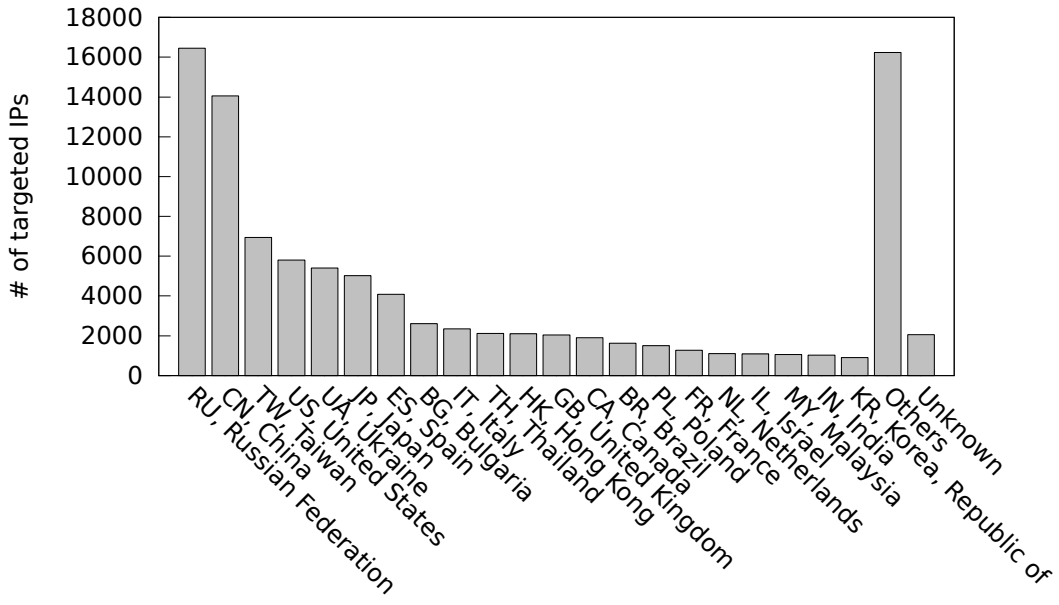
To provoke this behavior, we generated over 10000 random SHA-1 keys and searched for them in the DHT. If all DHT members are working correctly, it is to be expected that we do not receive any answers, as hitting a specific existing SHA-1 key by randomly generating it is highly unlikely (a chance of 1 to 2^{160} to randomly hit a specific existing key). However, on average 3.51 nodes answered our searches, resulting in about 9.79 invalid contacts per lookup. We used a geographic IP database[7] to get an overview which countries the wrongly acting nodes are replying (see Figure 3.14a) from, and to which country the returned IPs point (figure 3.14b). We can not determine, whether these wrong answers stem from broken clients or if they are intentional. As BT clients will try to establish a connection to returned IPs, a rationale for returning bogus values when receiving any query can be the intention of misusing the BT swarm to perform a DDoS attack against a specific target [59].

Due to the *DTNIQ*/*DTNIR* handshake this large amount of bogus answers is not a

[7] `http://www.maxmind.com/app/geolite`



(a) Location of the responding DHT node



(b) Location of returned results

Figure 3.14.: Bogus answers in the BitTorrent DHT

Duration [s]			Occurrences	Ratio
0.00	to	155.65	4	0.40%
155.65	to	191.30	20	2.00%
191.30	to	226.95	55	5.50%
226.95	to	262.60	151	15.10%
262.60	to	298.25	282	28.20%
298.25	to	333.90	269	26.90%
333.90	to	369.55	147	14.70%
369.55	to	405.20	54	5.40%
405.20	to	440.85	14	1.40%
440.85	to	476.50	1	0.10%
476.50	to	512.15	1	0.10%
512.15	to	547.80	1	0.10%
	>	547.80	1	0.10%

Table 3.2.: Duration of 1000 announcements

problem for nodes implementing BT-DHT: Sending a *DTNIQ_* is just a single UDP packet and does not incur much overhead. Only nodes which answered with a correct *DTNIR* will be announced to the DTN daemon.

3.5.7. IBR-DTN Implementation

As mentioned in the beginning of this section as of version 0.8 BT-DHT is integrated into IBR-DTN. The actual DHT implementation and the DTN specific extensions are located in a separate library[8] which is independent from the IBR-DTN code. That library can be used to integrate the DTN DHT into other DTN daemons. A patch to integrate BT-DHT functionality into DTN2 is available[9].

The integration of the library into IBR-DTN supports many configuration options to fine tune the performance. For bootstrapping (finding the first DHT neighbors), a DNS query can be used. Additional DNS locations for lookup can be configured. Alternatively, or in addition, IP/port information of DHT members can be specified directly. To allow faster bootstrapping a file can be used to store the last known DHT neighborhood persistently between daemon restarts. A node can opt to only query the DHT but not announce itself. Likewise, a node can decide whether it wants to publish its known neighbors to the DHT. There is also an option to opt-out from being published to the DHT by a neighbor. This wish is communicated through an additional IPND field, and thus, it is only honored when a neighbor was detected via IPND. All configuration options can be found in Appendix A.1.

[8] `https://github.com/ibrdtn/dtn-dht`
[9] `http://sourceforge.net/p/dtn/patches/5/`

3.5.8. DHT-based Naming Advantages and Challenges

The DHT-based naming mechanisms discussed above enables the BP to bridge the Internet with ease. Instead of devising new routing protocols which could cope with the scalability challenge the Internet provides, we utilize the existing routing mechanisms underlying the Internet and implemented a DHT-based naming service that can look up any EID and scales at least to millions of nodes as has been proven by various P2P applications utilizing the same approach. Thus, from the point of view of the BP, the whole Internet is seen as local neighborhood, where names can be looked up on-demand in contrast to the proactive beaconing of local-network mechanism such as IPND.

Especially considering mobile devices, the convergence layer information of a node might be looked up, without that node being available. With Nasdi we proposed an asynchronous notification mechanism that would alert the sending node, once the communication partner comes back online. If the bundle has not yet expired the sending node can deliver it. With the BT-DHT implementation we keep querying for a node from time to time until an entry is eventually found or the bundle times out, which basically achieves the same.

However, for battery-powered mobile devices such as smartphones neither continuous discovery nor maintaining constant contact with a DHT is practical. In the next section we will introduce a mechanism that allows such devices to be duty-cycled more aggressively while still maintaining good connectivity in a DTN.

3.6. Free DTN Routers: Mail Convergence Layer

As lined out in this chapter constantly joining and leaving the DHT is not suitable for mobile devices. From the DHT perspective high network churn negatively affects performance while from the devices' point of view the energy expended to keep in contact with the DHT might be prohibitive. Additionally, if both partners are mobile devices there is a chance that they will miss each other. In a DTN this is usually not a problem, if you transfer the bundle to other DTN routers which are available at the moment. Within the discussed DHT systems that would be proxy nodes which announced a disappeared node and are still reachable.

The question is how to get some always-on BP routers in the Internet: Who will operate them or pay for them, especially when BP market penetration is still low. Similarly, to the BT-DHT, where we piggy-backed an existing system, we found a solution for the shortage of public DTN routers in the Internet. The DTN community is always eager to point out that one of the oldest and most integral parts of contemporary Internet has always been based on DTN principles: The Internet mail system based on SMTP (Simple Mail Transfer Protocol) [60, 61]. Other classical Internet services with DTN qualities include UUCP (Unix to Unix Copy Protocol) [62] or NNTP (Network News Transfer Protocol) [63, 64], however SMTP is the service that is most widely used until today.

This notion has led to development of projects aiming to bring mail-based Internet connectivity to remote rural communities using a DTN [65, 66]. Here we will discuss the opposite approach: Using the Internet mail system to transport Bundles in a DTN by

designing a suitable Mail Convergence layer (MCL). Despite the obvious feasibility of such a scheme, it also provides advantages over other convergence layers and can qualitatively enhance the capabilities of a DTN, especially for mobile users as will be discussed in Section 3.6.4.

3.6.1. Requirements and Architecture

The goal is to use unmodified mailservers as DTN routers. This implies, that bundles can be sent to a mailserver and stored there until they are retrieved. The complete BP should be supported, including standard and custom blocks that might be encountered in a valid BP bundle.

DTN nodes will use SMTP to send bundles to other DTN nodes. The recipients mail server acts as an external bundle storage, until the receiver downloads its mails and extracts the contained bundles. The goal of the MCL's architecture was to enable easy parsing of bundles and to be compatible with standard e-mail servers. Therefore, we have chosen to encode the fields from the primary bundle block as well as fields from the payload block (processing flags, length) directly into mail headers, while the individual blocks will be put into separate MIME attachments. This design choice allows a BP implementation to fetch the mail headers first, and based on this information an early decision can be made whether the encoded bundle is of interest to the node or not. Furthermore, while not investigated in this work, this would allow some lightweight SIEVE-based [67] routing modules which can operate by just forwarding a mail without the need to parse or download the bundle.

An example of a bundle encoded using the MCL can be seen in Figure 3.15. It shows a bundle sent by a node with mail address *sender@server* and delivered to another node using the mail address *recv@server*. The EIDs are encoded as SMTP headers and the payload can be found in the attachment named "payload.data". The complete specification (see Appendix A.2) can be found in the Internet draft describing the MCL [68].

3.6.2. Propagation of Mail Addresses

While not relevant to the design of the MCL protocol itself, there needs to be a mechanism how a node learns about the mail address of another node. In addition to static configuration, the MCL implementation of IBR-DTN opts to extend the two standard discovery mechanisms: IPND [34] for local networks and BT-DHT [24] for Internet-scale node discovery. It seems that learning an e-mail address via IPND at first does not make sense, because once you discovered a node in the same subnet, a direct TCP connection is most likely the more efficient way to transfer bundles to that node. The rationale for this approach is that the IBR-DTN neighbor list will retain the information about the e-mail address longer than other discovery information. Thus, after having seen a node once in the local neighborhood, the mail address is known and can be used as a fallback if there are bundles available for that node, but the node itself is not. The same is true for e-mail address information extracted from the DHT. The BT-DHT has the additional advantage that for around 30 min after a node's departure from the DHT the stored information can still be retrieved [24]. Another possibility is a specific EID scheme such as *email://* that

```
Return-path: <sender@server>
Envelope-to: recv@server
Delivery-date: Wed, 23 Jan 2013 19:44:25 +0100
From: sender@server
To: recv@server
Subject: Bundle for mail://sender@server
Bundle-EMailCL-Version: 1
Bundle-Flags: 144
Bundle-Destination: dtn://some/eid
Bundle-Source: dtn://second/eid
Bundle-Report-To: dtn:none
Bundle-Custodian: dtn:none
Bundle-Creation-Time: 412281870
Bundle-Sequence-Number: 1
Bundle-Lifetime: 3600
Bundle-Payload-Flags: 8
Bundle-Payload-Block-Length: 35
Bundle-Payload-Data-Name: payload.data
Content-Type: multipart/mixed;
 boundary="=_f-20r0xUuORzjAo2CVz1bGFWJK1irHf4t+jNIoYURaTVkAY6"

This is a multi-part message in MIME format. Your mail reader does
not understand MIME message format.
--=_f-20r0xUuORzjAo2CVz1bGFWJK1irHf4t+jNIoYURaTVkAY6


--=_f-20r0xUuORzjAo2CVz1bGFWJK1irHf4t+jNIoYURaTVkAY6
Content-Type: application/octet-stream
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=payload.data

VGhvcm91Z2ggcmVhZGVyIGFjaGlldmVtZW50IHVubG9ja2VkIQ==
--=_f-20r0xUuORzjAo2CVz1bGFWJK1irHf4t+jNIoYURaTVkAY6--
```

Figure 3.15.: MCL-encoded bundle example

encodes the mail address directly.

3.6.3. Semantic Issues

Apart from the protocol issues there is the question, how to model the Internet mail system in terms of BP terminology. Is it considered (1) an extension of a node, and thus, of no relevance to the view of the network, is it – being implemented as a convergence layer – (2) just another communication channel such as TCP or IEEE 802.15.4 or is it (3) an independent node? To get a better grasp on this we compare it with the more standard TCP-CL. In the BP you have only EIDs identifying a destination (most often equaling a node). Mechanisms like IPND, DHT or static configuration can be used to map CL addresses (such as an IP) to an EID. Using the TCP-CL, a direct communication channel to the entity responsible for a given EID can be established. Bundles sent through the TCP-CL to a node can be considered delivered or forwarded. Depending on the bundle the receiving node might acknowledge reception or even take custody of the transferred bundle. This is not the case for the MCL. The receiver might not retrieve its bundle in time before it is expired, or the mail might simply get lost because the destination mailbox is full or non-existing due to outdated MCL addressing information. Therefore, marking a bundle as delivered when it is passed through the MCL to a destination's mail address might not be the optimal solution.

Our implementation opts to adopt the concept of a virtual node: When the mail address

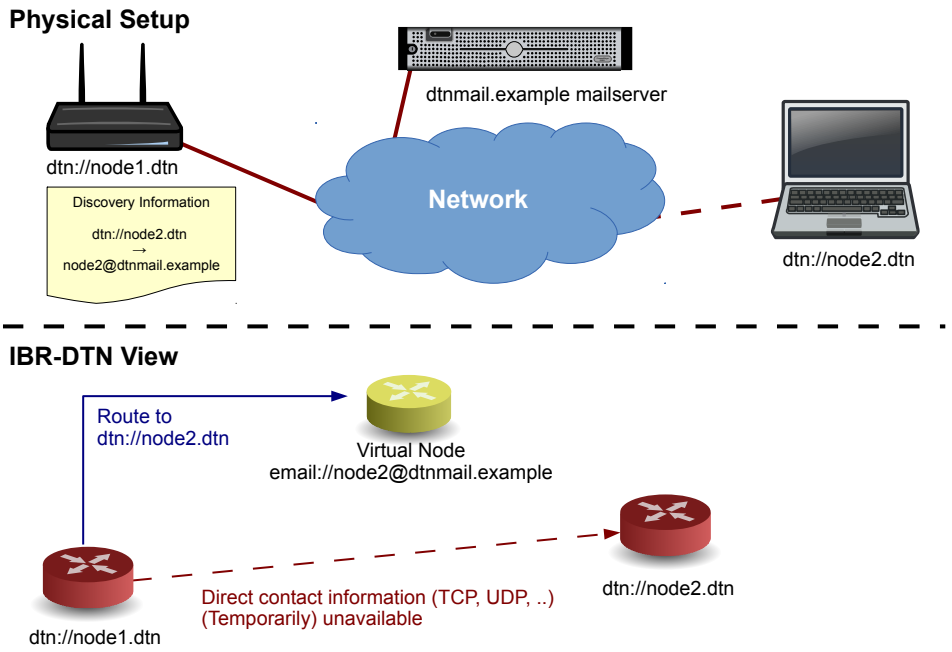


Figure 3.16.: Internal representation of MCL contacts in IBR-DTN

for a contact is available (through IPND, DHT or by receiving a bundle through the MCL), IBR-DTN adds a virtual node object representing the mail server and sets up a route to the destination through the virtual node. This situation is depicted in Figure 3.16, where node 1 has MCL contact information for the temporarily offline node 2. Using the convention of a virtual node, a bundle transferred through the MCL is not considered delivered, but only forwarded. After the destination receives the bundle from the IMAP server it can still send an ACK back or take custody if required, as these mechanisms also work across several hops in the BP. Furthermore, because the bundle is only forwarded, should the sender meet the destination later, before it had the opportunity to download the bundle from the mail server, the bundle will be delivered directly, thus ensuring the shortest possible delivery time. Once a node detects such an already received bundle on its IMAP server, it will delete that mail based on the header-information without wasting additional bandwidth by downloading it.

3.6.4. Usage Scenarios

While the MCL is not intended to replace the more traditional CLs such as TCP, there are two use cases where the usage of the MCL is very suitable: (1) Mobile scenarios that require asynchronous communication and (2) Interfacing legacy systems that are not able to run a complete BP framework.

Mobile Scenarios

Consider a mobile DTN consisting of nodes that can join or leave the network and can come into each other's range at any time. In such opportunistic networks, transmitting data to the destination can be challenging. We consider a scenario where node n_0 wants to

transfer a bundle to a previously encountered node n_1. n_1 will become available sometime after n_0 is already online. Figure 3.17 compares 4 different times A, B, C and D when n_0 will go offline and compares the effects on bundle transmission. Depending on the chosen discovery method and the time n_0 goes offline we will get the following results:

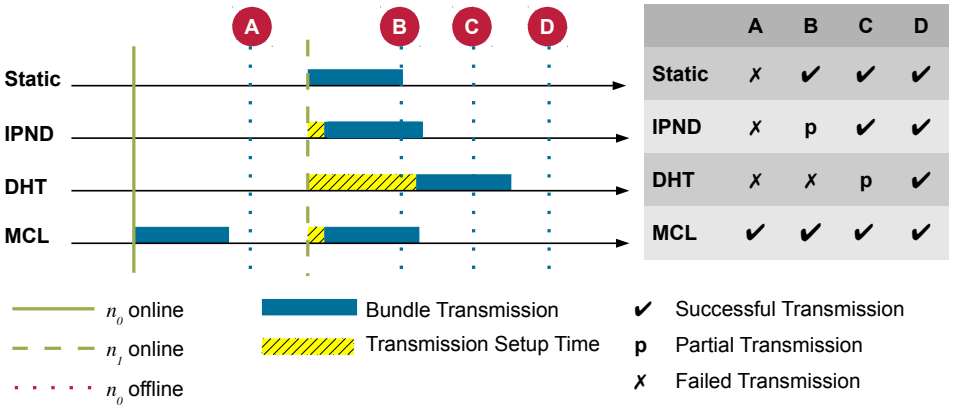
- **Static:** This is the baseline scenario. It is based on the unrealistic assumption that n_0 knows exactly at which time n_1 will be available. Such hard-scheduled contacts are usually only available in IPN scenarios. In this baseline scenario transmission is always successful, except in situation A, where n_0 leaves the network before n_1 joins.

- **IPND:** This is a very common scenario for opportunistic networks: Nodes employ some sort of discovery mechanism such as IPND [34], and once a suitable neighbor comes into communication range, data is exchanged. With any neighbor discovery mechanism there is a non-zero latency due to the trade-off between energy efficiency and latency (see Section 3.2). In the example in situation B the data can not be transmitted fully due to the additional overhead.

- **DHT:** For BP nodes with a stable Internet connection the BT-DHT naming service discussed in Section 3.5 is a good option for resolving EIDs. However, due to the high delays in building and joining the DHT, it is not the best choice in highly dynamic scenarios. In our example, once n_1 comes online it needs to associate itself with the DHT before it can announce its contact information, which can then subsequently be queried by n_0. This process can take up to several minutes (see Section 3.5.6). In our scenario situation C is the first one that allows for partial transmission of the data and only situation D allows successful completion of the transfer.

- **MCL:** The MCL is the only mechanism that enables complete transmission of the data in our scenario in all situations. The MCL approach is the only one that does not require any overlap in the online times of n_0 and n_1. As we assume n_0 already had contact to n_1 before, it knows n_1's MCL address. Therefore, n_0 can begin transmission to the mail server immediately, and the data transfer succeeds even in situations when n_0 leaves the network before n_1 joins it.

There are more conceivable options where using MCL is the best situation: Mobile devices in cellular network suffer from highly varying degrees of bandwidth. With highly asynchronous bandwidths, where n_0's uplink bandwidth is very high compared to n_1's download bandwidth, direct transmission rate is limited to n_1's lower bandwidth. If either of the devices can not stay online long enough, the data can not be fully transferred. Using the MCL n_0 can make full use of its bandwidth, and allow n_1 to download the bundle anytime later, without the need of n_0 being online. Of course, the same arguments holds, if n_0 and n_1 are never online at the same time: The asynchronicity of the MCL allows communication between the nodes, with the mail servers acting as "lightweight" DTN Routers.



	A	B	C	D
Static	✗	✔	✔	✔
IPND	✗	p	✔	✔
DHT	✗	✗	p	✔
MCL	✔	✔	✔	✔

Figure 3.17.: Bundle transmission between 2 intermittently connected nodes

Interfacing Legacy Applications

The MCL is also a great way to interface applications to a DTN without the need for a complete BP implementation: Simple MCL compliant bundles can be crafted by any scripting language, for example a simple PHP-based webapp can be used to generate a valid MCL encoded bundle without the need to install any extra libraries. Likewise, simple bundles could be parsed by applications not implementing the full BP specification.

When deploying a DTN system, it is quite likely that legacy applications need to be interfaced. Usually an application specific proxy incorporating a BP networking stack and application specific interfaces needs to be developed. When a full BP stack is not needed, you might opt to teach the legacy application a basic subset of the BP, just enough for the communication needs of the application. When developing the MCL implementation for IBR-DTN, we found the MIME-based MCL encoding to be much more accessible to third party tools than the normal on-wire format of the BP. For example, we whipped up a very small set of PHP functions that could run on any webserver supporting PHP, generating and parsing syntactically valid MCL bundles containing a single payload block. The generated bundles are fully MCL compliant and accepted by IBR-DTN. Using this code, a standard, low-cost shared webhosting account providing PHP can be used to interface a BP-based system without the need for any third party libraries that are usually not available on such machines.

3.6.5. Evaluation

As the MCL is reliant on adding various mail headers to an e-mail, we did test whether common mail providers will preserve the new headers when sending or receiving mail. We tested 7 popular free-mail providers by adding the extra headers to a mail sent to or received by them. To emulate some more complex e-mail processing we configured a forwarding chain including 5 separate mail providers[10]. In all these tests the injected mail headers have been preserved, which indicates that the MCL approach should work across

[10] Mails have been forwarded from Yahoo to Gmail to GMX to T-Online and finally to mail.ru with all custom headers intact.

most mail server setups.

All things being equal, the base performance of the MCL is somewhat lower than using the TCP-CL, as the necessary Base64 encoding [69] required by the MIME standard increases the transfer volume by around 33%. Generally, to minimize the transfer volume it is a good idea to compress the contents of a bundle. For this purpose IBR-DTN supports the transparent compression of bundle payloads indicated by an additional Block. This is however not standardized in RFC 5050 [5] and thus not compatible with other BP implementations. In the following experiments the bundle size is always the payload size of the bundle before it is encoded by the MCL and no further compression is applied. We performed two experiments to give an idea what kind of bandwidth and latency can be expected when using the MCL with common mail providers.

Throughput

We measured the throughput that can be achieved using common e-mail providers. In this test node n_0 sends a packet to n_1 using the MCL. n_1 has been configured to use different free-mail providers, while n_0 always used the same outgoing mailserver, which was under our control. We set the bundle size to 1, 5 and 10 MiB, and performed 100 transmissions for each bundle size and mail provider. The results are shown in Figure 3.18. We conclude that generally larger bundles offer better bandwidth. The observed bandwidths range from 614 $kBit/s$ for T-Online and 1 MiB bundles to 5738 $kBit/s$ using freenet with 10 MiB bundles. Several factors contribute to the total observed bandwidth: In addition to the SMTP and IMAP transfer speeds there is always a fixed per-bundle overhead for initiating the connection to the SMTP and IMAP server and some lost time due to the polling interval, which was set to 5 seconds for this experiment (the IMAP library used in our implementation does not yet support the IMAP IDLE feature). Many free-mail providers limit the maximum attachment size, so sending arbitrarily large bundles is not possible (however, BP fragmentation can be used). Interestingly, Gmail shows consistently slower performance for 10 MiB bundles than for 5 MiB bundles.

The results shown in Figure 3.18 are the total bandwidth from n_0 to n_1. Keep in mind that transferring a bundle using the MCL includes at least two, more likely three, separate sequential transfers: First the sender has to upload the data to his mail server using SMTP (t_{s_in}). If the sender and receiver use different mail providers, the sender's mail server needs to forward the mail to the receiver's SMTP server ($t_{s\_forward}$). After reception of a mail, before delivering it to a user's mailbox there will be some additional processing such as malware scans or antispam filtering (t_{proc}). After the mail has been delivered to the receiver's mailbox, it needs to be downloaded using IMAP (t_i). Due to the nature of the Internet mail system all these transfers are sequential for one bundle. Additional delays are incurred by the processing and polling intervals of the MCL implementation (t_{MCL}). This results in the total time needed to transmit a bundle using the MCL as $t_{total} = t_{s\_in} + t_{s\_forward} + t_{proc} + t_i + t_{MCL}$. The breakdown of the total time required when transmitting a bundle using the MCL is also shown in Figure 3.19.

An interesting case is Gmail, where the bandwidth seems to decrease for larger bundles.

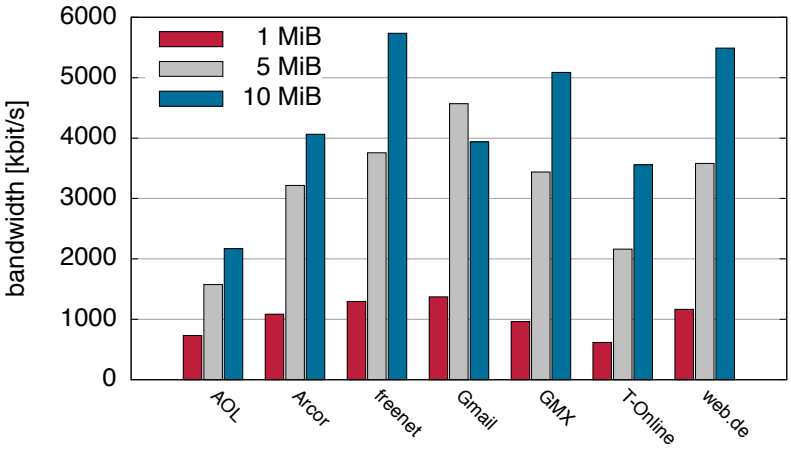


Figure 3.18.: MCL throughput with different freemail providers

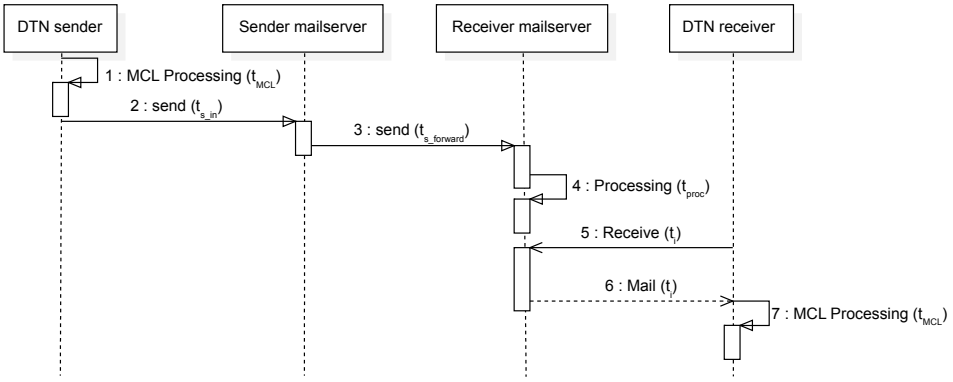


Figure 3.19.: MCL processing and transmission time breakup

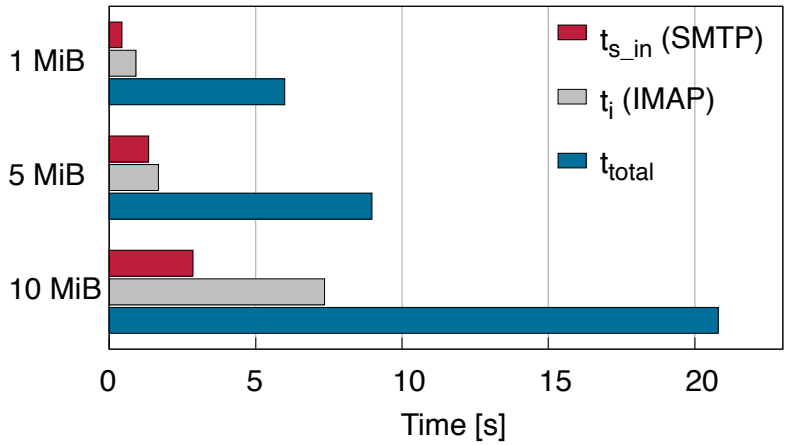


Figure 3.20.: SMTP and IMAP transfer times using Gmail

Figure 3.20 shows t_{s_in}, t_i as well as t_{total} for transferring bundles from our mail server to a MCL node using Gmail. The displayed values are the average of transferring 100 bundles. Obviously for t_{total} it holds $t_{total} > t_{s\_in} + t_i$. For 1 MiB bundles the SMTP transfer takes 427 ms and the IMAP transfer from Gmail takes 903 ms. In this case t_{total} is $\approx 5.98\ s$. In this case the overhead $t_{s\_forward} + t_{proc} + t_{MCL}$ is $\approx 4.65\ s$. For 5 and 10 MiB bundles the overhead is $\approx 5.95\ s$ and $\approx 10.60\ s$ respectively. The increase in overhead for 10 MiB bundles is larger than expected. This is in line with the results from Figure 3.18, where Gmail's performance suffers with larger bundles. Another factor limiting the achieved bandwidth when transferring larger bundles using Gmail is lower IMAP bandwidth for large mails. While 5 MiB mails could be downloaded with $\approx 24.5\ Mbit/s$ on average that speed plummeted to $\approx 11.1 Mbit/s$ for 10 MiB bundles.

While t_{total} might seem high, and the overall achieved bandwidth seems low, keep in mind that the sender will only see t_{s_in}. It can forward a bundle quickly. Similarly, a receiver only sees t_i: From its viewpoint the other components are just delays the bundle has accumulated in the DTN. Furthermore, in our MCL implementation the receiver will receive a bundle already forwarded through the MCL directly *if* the source is available, before it is downloading it from the IMAP server. As described in Section 3.6.3, a bundle delivered to an MCL-enabled mailbox is only considered forwarded, not delivered. If the destination node is encountered in a direct contact later, and it has not yet downloaded the bundle from its mailserver, the bundle will be forwarded and delivered directly. Thus, *if* a bundle is received through the MCL it *was* the fastest possible path through the DTN. To get a better understanding of $t_{s\_forward}$ and t_{proc}, the following section will take a look at the latencies that can be achieved using the MCL.

Latency

In this test we measured the latency that can be achieved between two nodes, n_0 and n_1, using the MCL. Node n_0 always connects to our own local mail server, while n_1 uses various free-mail providers. The `dtnping` tool was used to send 25 ping probes from n_0 to n_1 (the

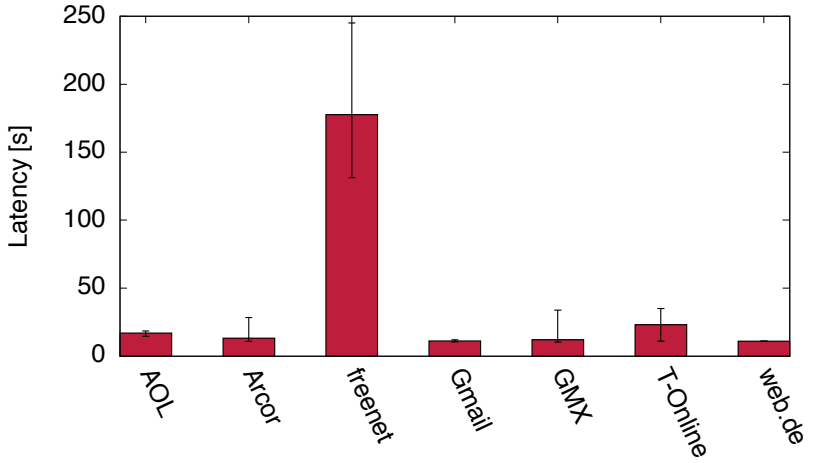


Figure 3.21.: MCL latency with different freemail providers

most restrictive free-mail provider used for this test limited the number of mails that could be sent in a row to 25). Latencies consist of the latencies introduced by the MCL implementation and the processing time at the mail servers. The most relevant factor for the MCL overhead is polling time: As the used IMAP library does not yet support the IMAP IDLE extension, we opted for a poll interval of 5 seconds in this experiment. Another delay comes from an optimization in our MCL implementation: To avoid repeated connections to a SMTP server, newly generated bundles will be enqueued for up to 10 seconds, before the mailserver is contacted and the bundles are transmitted. As these conditions are the same for every run, the measured data shows speed differences in the mail providers processing.

The results for this experiment can be seen in Figure 3.21. The average latency values range from 11 *s* for Gmail and web.de to 177 *s* for freenet. The maximal observed time is 240 *s* (also freenet). The results demonstrate, that with common freemail providers real-world latencies between 10 to 25 *s* can be expected. However, as different mail providers optimize their systems in different ways, and latencies are a very secondary concern in mail systems, delays can be much higher, as can be seen with freenet. As expected, the MCL is no fit for real-time applications, but with many providers the achieved latencies still allow using messaging applications without decreasing user comfort too much.

3.7. Summary

In this chapter we took a look at approaches that allow DTN applications to scale comparably to applications based on classical Internet technologies. We saw that established DTN approaches for node discovery such as beaconing and flooding-based multiple-copy routing schemes do not scale to large networks. As routing based on IP addresses is the standard case in the Internet, and is proven to be working, we argued that a special DTN-based routing in the Internet is not required, but instead you want to have a reliable

mapping of EIDs to IP addresses and let the IP-based convergence layers handle the rest, basically treating the whole Internet as local neighborhood.

To map the unstructured EID name space to IP addresses, we proposed using a DHT. This data-structure is well-suited for unstructured data. It has already been proven to work in many P2P applications and does not need additional infrastructure. We presented a DHT structure that is well-adapted for DTNs supporting an advanced notification mechanism to make use of contacts as soon as they are available. As practically speaking the BP is not yet wide-spread enough to allow for a stable and resilient DHT-based on BP nodes, we implemented and adapted the DHT approach using the BT-DHT. This is in line with our goal of using as much of the available infrastructure as possible: The BT-DHT is already well established consisting of a great number of nodes. By adapting and extending it without breaking compatibility with existing BT clients, DTN applications can use a stable DHT to resolve EIDs to IP addresses immediately while strengthening the BT-DHT.

Even with discovery and routing solved, for mobile devices there is the additional problem of asynchronicity: While more and more mobile devices have Internet connectivity and can support BP applications, we argued that direct exchange between two mobile devices is unlikely. The Internet connection of mobile devices is often duty-cycled for energy efficiency reasons and for the same reason most local discovery mechanisms are not a good idea for battery-powered devices. In a DTN this means some always-on DTN routers are needed that can bridge the gap between mobile devices with different sleeping schedules. We have shown that, instead of deploying stand-alone DTN routers, this use-case can be equally well solved by leveraging the Internet mail system using a special convergence layer. Again the capabilities of an Internet-based DTN system are extended, without the need for further infrastructure.

It is possible to deploy Internet-wide DTN applications today. Software needs to be extended on the end-systems, however intermediate systems do not need to be touched. Existing and proven Internet technologies can be used to extend the capabilities and performance of a DTN.

4 Storage Synchronization

4.1. Problem Statement

As we have seen in the previous chapter, when applying the paradigm of DTN to the Internet there is often the problem of scale: Broadcast-based discovery did not work, and replicating routing mechanisms are not applicable at all. There is another scaling challenge, inherent to the DTN paradigm: Data volume. While the presented DHT naming service and the Mail convergence layer allow a DTN application to effortlessly bridge the complete Internet to send bundles to an arbitrary back-end, lets consider the case of large content and telecommunication providers: The store-carry-and-forward architecture of a DTN lends itself very well to caching, as every node has a bundle storage. Consider a video provider such as Youtube which hosts billions of individual video clips. Even the English Wikipedia already contains around 4.5 million articles[1], each of them consisting of several objects such as images. When such contents are distributed throughout the DTN they will get replicated, either implicitly by means of a routing protocol or explicitly. Today Youtube is using a similar idea, when it is placing caches in ISP's networks to lessen network load and increase users' performance [70]. This problem is only exacerbated with the dynamic nature of today's Internet services: In 2013 every day 4.75 billion items have been shared and more than 10 billion messages have been sent on Facebook alone [71]. According to data from 2014 [72] every *minute* 10 million WeChat messages are sent, 38,194 photos uploaded to Instagram and 138,889 hours of video are watched on Youtube.

The challenge here is not the amount of storage: The prices for storage continue to decrease, and every participant in the system can find its own trade-off between investment in storage capacity and network load. A major problem is how to keep those caches of a large number of distinct objects in sync. It turns out that the basic question: "Which objects do I have, which the other node does not have?" is not easy to answer in a computationally and bandwidth efficient way. Even with advanced, selective replication techniques, which replicate data according to some priorities or other decision criteria, this basic problem needs to be solved first.

Parts of the work presented in this chapter have already been discussed in [73].

4.2. The Synchronization Problem

More formally for two collections A and B residing on the respective DTN nodes E_a and E_b the following information needs to be available:

Node E_b needs to know the set $A \setminus B$ and E_b needs to know $B \setminus A$ for a push scenario. For

[1] data from 2014, `http://www.wikipedia.org`

⇓	**Pull scenario**	**Push scenario**
Required information at node E_a	$B \setminus A$	$A \setminus B$
Required information at node E_b	$A \setminus B$	$B \setminus A$

Table 4.1.: Required information for synchronizing two sets

a pull scenario b needs to know $B \setminus A$ and E_b needs knowledge about the contents of $A \setminus B$ (see Table 4.1). The question, whether a given element x is in set A, is a membership query.

While this amounts to a simple list comparison for small datasets it is not feasible for really large collections because the overhead of transferring complete lists is too large compared to the available bandwidth and the size of a single data item. As an example a static HTML dump of the English Wikipedia without pictures consists of 233 GiB in 14.257.487 files. The whole collection can be compressed to around 15 GiB[2], which shows that it is reasonable to expect similar amounts of data on mobile consumer devices such as phones or tablets in the near future. The average path-length of each file is ≈ 50 bytes. Exchanging lists of the file names would thus lead to around 680 MiB transferred data for the complete collection. Even when identifying items through an 20 Byte SHA-1 hash, this still leads to a list of size 271 MiB.

Whenever it is prohibitive to exchange lists due to size constraints, Bloom filters are a widely used alternative data structure [74]. A Bloom filter is a very efficient probabilistic data structure which allows for membership queries. A Bloom filter is very small compared to the list at the cost of an arbitrarily small false positive rate. Bloom-Filters were introduced by Burton Bloom in 1970 [75]. A Bloom filter for a given set can be very small. Unfortunately, Bloom filters exhibit a false positive rate, i.e. sometimes the answer to the question, whether an item x is in a set A is positive, even though there is no such element in set A. Therefore, Bloom filters can only be used in applications which can tolerate a small error rate. We will introduce Bloom filters in Section 4.4. A prominent use-case for Bloom filters is caching [76, 77]: A Bloom filter is used to determine whether a cache contains a certain item. When the Bloom filter answers positive, the cache is queried for that resource. If the Bloom filter's answer was a false positive and the cache does not contain the item, the original source is asked. However, for many scenarios, including the DTN use-case we are interested in this thesis, a plain Bloom filter is not acceptable, as its false positive rate would mean, some elements could never be synced. What one would really like is the space efficiency of the Bloom filter without the false positives. Therefore, for the approach discussed here, we will start with a Bloom filter, and subsequently deal with the false positives.

Many Bloom filter variants and extensions have been proposed. Counting Bloom filters [76] add the ability to delete elements from a filter. The requirement that the number of elements to be added needs to be known before the Bloom filter is created, has been relaxed by compressing sparsely populated filters [78], or by allowing a filter to grow with

[2] Compressing all files into a single archive using LZMA

demand [79]. While these and other variants all improve certain aspects of plain Bloom filters, they all still have the fundamental issue of false positives. While in caching applications any false positive leads just to slightly higher overhead for corner cases, false positives cannot be tolerated for synchronization use cases such as file replication where the goal is to have two identical sets with no missing items after the process. Using a hash trie-based data structure in combination with Bloom filters has been discussed in [80, 81]. However, there the goal was to achieve set reconciliation using only a single round-trip, which only reduces false positives compared to a vanilla Bloom filter, not eliminating them. Therefore, this approach has fundamentally the same limitations as other Bloom filter variants: While it improves the amount of false positives, it cannot be used for applications that can not tolerate them.

A computationally complex approach based on representing sets as polynomials and analysis of lower bounds of communication complexity for set reconciliation has been presented in [82]. In [83] Eppstein et al. suggest using an approach based on Invertible Bloom Filters (IBFs) [84] to tackle the set reconciliation problem. Instead of setting elements in a Bloom filter to 1, an IBF uses the XOR operation combined with a counting field. This makes it possible to reverse additions to the Bloom filter, if one of the hashes for an element only has a count of 1. In order to parameterize the IBF correctly, the difference set's size needs to be estimated beforehand. Estimating the difference set's size is non-trivial and there is a probability that, depending on the estimation and the elements in a set, it may not be possible to decode an IBF recursively (i.e. there is no element in the Bloom filter that has only been set once). Therefore, this method only works with a certain probability and upon failure will require a second round using a larger IBF. Another approximate set reconciliation method also based on the difficult problem of estimating the difference set's size accurately has been presented in [85]. The authors state that they are particularly interested in optimizing the communicating overhead for almost similar sets. By not only estimating the size of the total difference set but also the size for each node separately their Bloom filters can be sized optimally.

Overall it can be said that mechanisms which are able to achieve the best communication complexity, can often only do so, if some crucial parameters are known or estimated with a high degree of precision. A failure to determine those parameters correctly leads to significantly worse performance or prevents the synchronization from finishing successfully. A simple Bloom filter on the other hand is quite robust to non-optimal input parameters. For reliable synchronization in dynamic DTN scenarios we require a set reconciliation method, that is as robust as a Bloom filter, computationally efficient and not suffering from false positives.

4.3. Assumptions and Conventions

In this chapter a single capital letter like X denotes a set, with $|X|$ being the cardinality of that set. E_X denotes the entity (the node, computer, system, ...) which is in possession of set X. B_X is the Bloom filter created from the contents of set X, T_X is the hash trie containing

all elements from X.

To avoid confusion for the remainder of this chapter we will use the term *node* only when referring to nodes within a trie. When referring to the systems which possess the sets and communicate for synchronization we will use the term *entity*. To synchronize two sets A and B, E_A and E_B need to exchange information. The amount of information that needs to be exchanged to decide which actual elements need to be transferred to equalize the sets, is used as a measure for the efficiency of a synchronization approach. When calculating the communication overhead, we do not consider overhead from lower level protocols that might be used in a concrete implementation. We assume byte level granularity of the communication channel, i.e. if we need to transmit 1 bit in a single message, we count it as 1 byte.

4.3.1. Exact synchronization

When we say our approach is false positive free, this is based on some assumptions: We assume that the elements in both sets are unique numbers. In applications these numbers usually will be identifiers representing some items. How to map items to unique numbers is application-specific, however we assume for practical applications often using hash values, generated by a collision free hash function, will be a suitable approximation of representing data as unique numbers. This makes the presented algorithm applicable to any application that is able to represent its data as hashes.

Independent from the chosen item representation our algorithm uses hashing to identify subsets. This fundamentally makes the system susceptible to hash collisions. For the evaluation we used SHA-1 as a hash function, generating identifiers of length 20 bytes. This approach is in line with the analysis of the set reconciliation method presented in [83], which only uses 4 byte identifiers and 4 byte hashes to safeguard against collisions from sums of permuted elements. So more formally speaking the approach presented here reduces the false positive rate of a Bloom filter, which might be in the area of a few per mill or percent down to the probability of hash collisions of the employed hash, which is many orders of magnitudes smaller (1 to 2^{160} for two arbitrarily chosen items using SHA-1) with negligible overhead. Applications needing even higher security guarantees can probably not rely on identifiers at all. Any advanced approach presented here or in related work would not be acceptable. Instead, such applications would require a bit-by-bit comparison of all their data for an "exact" synchronization. The approach of identifying arbitrary data as a unique hash is already used in real world applications such as BitTorrent applications, commercial backup solutions or the widely used *git*[3] version control system, which is also using SHA-1 and provides no safeguards against hash collisions.

4.4. Bloom Filter Primer

A Bloom filter [75] is a bit array of length m. All bits are set to 0 initially. To add an item x to a Bloom filter, x is hashed with k different hash functions $h_i, 1 \leq i \leq k$ with $0 < h_i(x) \leq m$.

[3] `http://git-scm.com/`

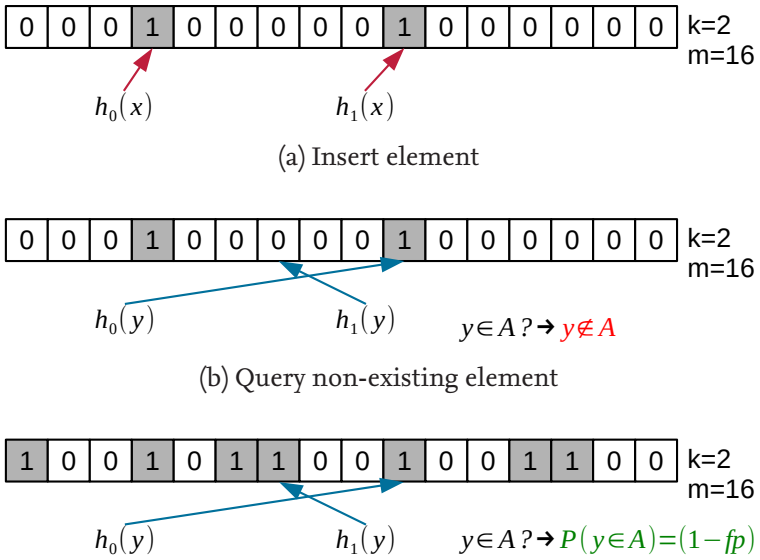


Figure 4.1.: Bloom filter operations

A bit is set to 1 in the filter for all $h_i(x)$. An example of adding an element to an empty Bloom filter can be seen in Figure 4.1a.

To check whether an item y is in set B the Bloom filter B_B of set B is needed. The item y is hashed with every function h_i used to construct B_B. Every position $h_i(y)$ in B_B is checked: If any of the positions is 0, one can be sure that y is not part of collection B. An example is shown in Figure 4.1b. However, if all $h_i(y)$ in B_B are set to 1, all that can be said is that y *probably* is in B (see Figure 4.1c). If $y \in B$, all bits $h_i(y)$ would be set to 1, but there is also the possibility that other items in B set those bits to 1 (hash collisions). Thus, the Bloom filter would yield a *false positive* for y.

Further analysis shows (see e.g. [74]) that the probability of a false positive when querying an element is approximately

$$\left(1 - e^{-kn/m}\right)^k \tag{4.1}$$

with

- k = number of hash functions
- n = number of distinct elements added to the Bloom filter
- m = size of the Bloom filter

As can be seen the false positive probability can be reduced using a greater m, because this means the hash space becomes larger and thus hash values are less likely to collide. Also decreasing n reduces false positives as fewer elements mean fewer hash values in the filter and thus less collisions. Regarding the number of hash functions k it is not quite

as easy: Using more hash functions *reduces* the number of false positives because k hash collisions are needed for a false positive. But on the other hand more hash functions also *increase* the probability of false positives, because the filter becomes more crowded, i.e. it is easier to "hit" an 1 in the filter. It can be shown [74] that the false positive probability with respect to k is minimized when

$$k = ln\, 2 \cdot \left(\frac{m}{n}\right) \tag{4.2}$$

This is the case, when half the bits in the filter are set to 1. Intuitively this makes sense, since in such a configuration the entropy of the filter is maximized.

4.5. Synchronization Approach

To reconcile two sets, communication partners need to know some meta information from the respective other set in order to determine which elements compromise the symmetric difference set. In this section we present three basic techniques to obtain this knowledge. We then introduce our approach, combining two of the basic techniques to achieve reliable synchronization and high efficiency.

4.5.1. Naïve approach: Exchanging lists

If two entities connect to synchronize two sets, the naïve approach is to exchange lists. That is, to synchronize A and B, E_B sends a list containing a hash for every item in B to E_A. The communication overhead of such an approach is linearly dependent on the size of the collection, $O(|B|)$. While this is a good solution for small datasets, it is not feasible for really large sets because the overhead of transferring complete lists is too high. As we have shown in the introduction, transferring the SHA-1 hashes of every item in the Wikipedia dump requires $\approx 271\ MiB$. Even when assuming that each article is an atomic unit, a list of hashes for all 4.5 million articles of the English Wikipedia still results in a $\approx 85.8\ MiB$ list. This lists need to be transferred even if no item is missing on the other entity. In a situation where DTN nodes possessing a large amount of bundles meet opportunistically this approach is not feasible.

4.5.2. Bloom filter with hash index

As we have seen, a Bloom filter for a set can be significantly smaller than a list of items. Given a Bloom filter B_B, the time-complexity of checking every element in A for membership in B is $O(|A|)$ under the assumption that hashing of individual elements in A is possible in $O(1)$ time. This can be achieved by using cached hash values for all elements in A: E_A can keep the hash value of each item as metadata (this also helps with deletion of elements from the Bloom filter). This guarantees $O(|A|)$ time when checking every element for membership. On a practical side this also means that each item has to be hashed only once when added to a collection. This is an advantage for resource and energy constrained devices.

Storing hashes allows for further optimization of the synchronization times. Remember,

given two collections A and B, our goal is to find those elements in A which are not already present in B: $A \setminus B$. Given two Bloom filters B_A and B_B it is easy to calculate a bit vector describing the hashes in the symmetric difference set of A and B by XOR'ing the filters: $B_A \oplus B_B$. This vector contains the hashes of elements either of the sets possesses exclusively. AND'ing this to B_A:

$$missing_B = (B_A \oplus B_B) \wedge B_A \tag{4.3}$$

results in a bit vector containing the hashes of elements in A, but not in B. Due to the probabilistic nature of Bloom filters the best result we can achieve is getting the hashes of all elements in $A \setminus (B \cup Fp)$, that is all elements which A contains except those that B does and except those we wrongly assume to be in set B due to false positives in B_B.

Assuming the Bloom filter size as constant this operation needs $O(1)$ time. Normally, this vector would be of little interest because we can not map hash values to items in our set, as this would imply reversing the hash. However, since we decided to store the hashes as metadata, it is trivial to build an inverted index mapping the stored hash values to set items. With this index it is possible to find all items corresponding to each marked hash in the resulting vector. Using binary search over the stored hash values, the items corresponding to a hash can be found in $O(log\ |A|)$ time. The upper bound for bits set in the vector is the number of elements in the set $A \setminus (B \cup Fp)$ multiplied by the number of hash functions k. Thus the time needed to retrieve all elements belonging to the set $A \setminus (B \cup Fp)$ is

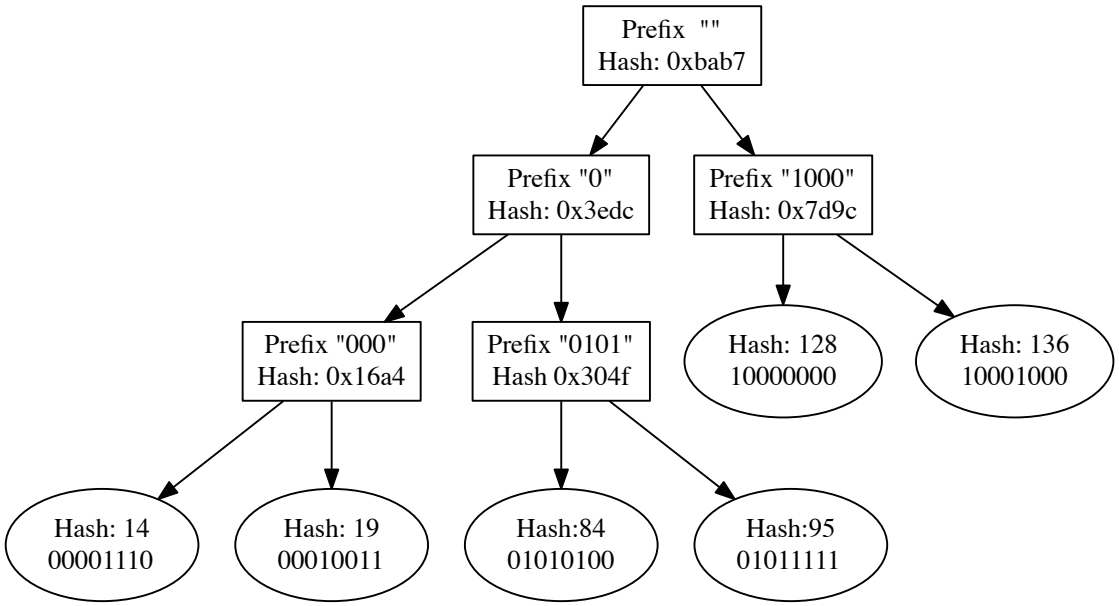
$$O\left(|A \setminus (B \cup Fp)| \cdot k \cdot log\ n\right) \tag{4.4}$$

Thus, overall the Bloom filter is very efficient when the mapping between hashes and elements is cached in a database. However, the remaining false positives might not be acceptable. Larger Bloom filters will reduce the false positives, but for asymptotically smaller gains the filter will grow exponentially larger.

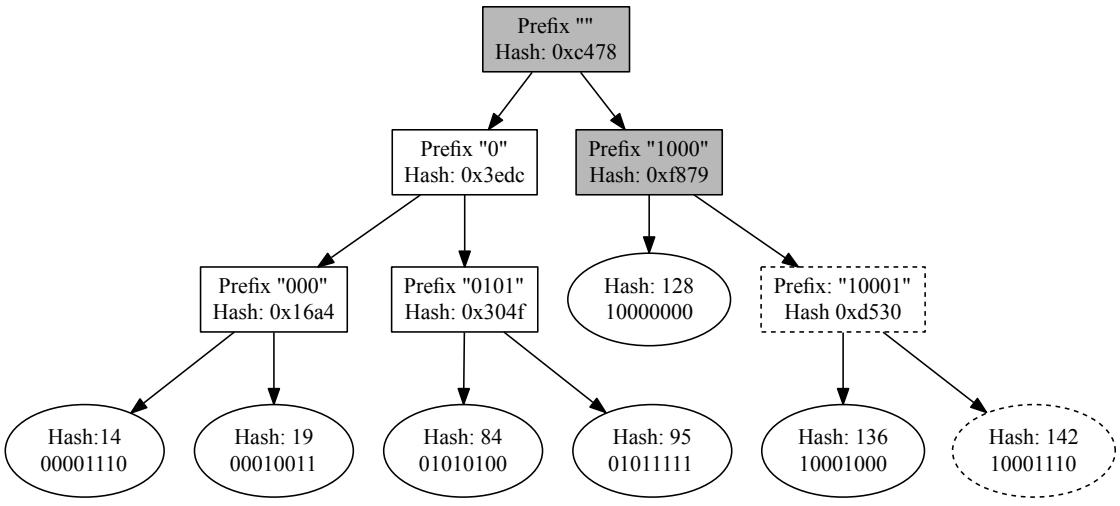
4.5.3. Hash tries

This approach uses a binary radix trie [86] of hashes as metadata to aid in the synchronization process. This data structure is also known as Merkle trie[87]. The basic idea is like this: If a sufficiently long good quality hash for a single element unambiguously identifies an element in a set, this is also true for a hash identifying a specific subset of a set. Therefore, instead of comparing hashes of individual elements, as in the naïve list approach, E_A and E_B can also compare hashes of subsets. The crucial thing is, that E_A and E_B need an approach to derive the same subsets. This is possible if we impose an order on the elements. As we represent items as hash values this is trivially done by ordering the items according to their hashes.

The hash values for all elements can be inserted into a binary trie. For example using 160 bit hash values the resulting trie has a maximum depth 160, with a hash value of 000...000 being the leftmost leaf and 111...111 being the rightmost leaf.



(a) Original trie



(b) Trie after inserting 1 element

Figure 4.2.: Propagation of changes in the trie

Figure 4.2a shows an example trie with 8 bit hashes. When traversing down from a node through the trie, the concatenation of the upper values of all nodes in the path from the root to the node is the prefix of all hash values that are in the leaves of the subtrie originating at that node. For example, the hash 19 can be found by traversing from the root to the leaf according to its binary representation: $0b00010011$. Each inner node of the trie holds a hash calculated by feeding its children hashes to the used hash function, e.g. the node 0101 contains a hash including information about all items which share the prefix $0b0101$ (84 and 95).

If E_A and E_B build a trie of their elements and calculate the combined hash values for intermediate nodes, corresponding nodes (nodes with the same prefix), can be compared: If the hash value for a pair of nodes representing the same prefix in both tries is equal, one can be sure (due to the collision-freeness of the employed hash function), that both collections contain exactly the same items in the leaves of the subtrie rooted at the compared nodes. Therefore, there is no longer a need to compare any nodes in the subtrie of a matching node – the search space can be pruned.

The problem is, that a difference in the leaves propagates all the way to the root node, i.e. the more dissimilar the collections are, the higher the probability that hashes in the first layers of the trie are dissimilar. This means, in many cases, the hash trie will perform worse than the simple list comparison, as in addition to all leaf hashes also the nodes encountered while traversing the trie have to be sent to the other entity. For completely dissimilar sets of size n the efficiency would degrade to $O(2n)$, which is worse than using a list. The impact of changing the set can be seen in Figure 4.2b where 1 item has been added to the collection from Figure 4.2a. The dashed nodes have been added to the trie, while the grey nodes are existing nodes whose hashes needed to be changed to reflect the new additions.

As the hash trie approach can prune whole subsets from the synchronization process by comparing a single hash value, it can be very efficient. However, due to the propagation of a single change through the whole trie, the set comparison is very inefficient if there are too many dissimilarities in the sets.

4.5.4. Combined Approach

As outlined in Section 4.5.2, using only Bloom filter information for synchronization can lead to incomplete synchronization due to false positives. The hash trie approach does not make such errors. However, due to the propagation of a single change through the whole trie, comparing collections based on hash tries can be quite inefficient.

Therefore, we propose to use a hash trie-based synchronization after a Bloom filter approach declares two collections similar. After the Bloom filter synchronization does not find new items to sync, the two sets are similar, *except* for the false positives. When choosing a sufficient small false positive rate for the Bloom filter, the sets either are

- Equal: Thus all corresponding nodes in both tries contain the same hash, and the whole trie can be pruned by only comparing the root node.

- Almost equal: The differences lie only in the false positives, which the Bloom filter could not detect. In this case it is to be expected that large subtries can be pruned when performing the hash trie-based synchronization, thus yielding an acceptable performance.

The combined approach is shown in Algorithm 3. Here E_A will push missing items to E_B. After B_B has been received, the hashes of the items missing on E_B are calculated in line 3. As we are using an index, the hashes are enough to determine the missing items and sent them to E_B (line 5). Next, the Bloom filter is used to find elements missing on E_A. E_A calculates the bit-vector containing hashes of the missing items ($miss_A$). In line 8 E_A decides, whether B_A or $miss_A$ is smaller when compressed, as E_B can use both information to determine which items to send. To save bandwidth, E_A will send the smaller compressed bit-vector to E_B.

After the Bloom filter phase the trie-based synchronization is initiated, to fix false positives (line 14). As the hash trie synchronization also needs to run both ways to ensure complete synchronization, in the last step in line 15 E_B is asked to start the trie-based phase.

Algorithm 3 Combined synchronization algorithm

Executed on E_A

```
1:  procedure SYNCHRONIZE(E_B)
2:      B_B ← RECEIVEBLOOMFILTER(E_B)
3:      miss_B = (B_A ⊕ B_B) ∧ B_A
4:
5:      SENDFILESWITHHASHES(miss_B)          ▷ Send files identified by Bloom filter
6:
7:      miss_A = (B_A ⊕ B_B) ∧ B_B                          ▷ Request missing files
8:      if COMPRESS(miss_A) ≤ COMPRESS(B_A) then
9:          REQUESTMISSINGITEMSM(E_B, miss_A)
10:     else
11:         REQUESTMISSINGITEMSBF(E_B, B_A)
12:     end if
13:
14:     WCBFS_TRAVERSE(T_A.root)          ▷ Use hash trie for remaining false positives
15:     REQUEST_WCBFS_TRAVERSE(E_B)
16: end procedure
```

4.5.5. Trie traversal scheme

For practical applications, the question is how to traverse the tree. A breadth first search (BFS) will prune subtries early. A depth first traversal will find the first different leaves fast and allows the application to transmit actual data early. Thus, performing a depth first search (DFS) is desirable to increase the goodput if the time available of communication is

short, which is often the case for opportunistic networking scenarios. User data can be transmitted, even before all the metadata necessary for synchronizing the whole sets has been exchanged.

Another practical consideration is the round-trip time of the used network between E_A and E_B: In the hash trie synchronization the position and the hash value of a node are sent to the other party. Depending on the answer, the subtrie will be pruned (hash similar), or the algorithm needs to traverse that node (hashes dissimilar, or node not existing in other trie). However, this would generate one round-trip for each node that is compared in the tries, and thus add the latencies for the round-trip as well as overhead from the used transport protocol to each compared node. Therefore, from this perspective a BFS, that could transmit all nodes on a trie level in a single message, is more desirable.

To balance between these two requirements, we use a width constrained breadth first traversal (WC-BFS): When traversing a node for which the hashes are dissimilar, all children of that nodes are candidates to be sent to the other entity. For the WC-BFS we define a target threshold, called *collect*, which is the desired amount of nodes we want to send to another entity for checking in one batch. The traversal algorithm is shown in Algorithm 4. The CheckNodes(*to_check*) call in line 2 is supposed to transmit the hash values and prefix of the nodes in the *to_check* set to the other entity, and upon receiving the answer return all nodes from *to_check* that have different hashes or are not existing in T_B. Children of dissimilar non-leaf nodes will be added to the returned *cand* set (line 8), dissimilar leaf nodes (when the other entity does not have the leaf asked for), will be transmitted to the other entity (line 6). At most *collect* items will be sent at once, so if the number of available candidates is larger, candidates exceeding the threshold will be put on the *stack* (line 12 - 14). If the number of elements in *cand* is smaller than *collect*, *cand* gets filled up with elements from the stack, as long as the stack is not empty (lines 16-18). Thus, the algorithm tries to always send *collect* nodes to the other entity for comparison. It might send less, but it will never send more. If there are no more candidates to send (line 20), the algorithm terminates. Using a stack biases the search towards lower levels of the trie, when the list of candidates is filled with values popped from the stack. An alternative would be using a FIFO, which would bias the search towards covering more breadth, when it runs out of candidates from the previous cycle.

4.6. Evaluation

4.6.1. Hash Trie Performance

For evaluation purposes the method described in Section 4.5.4 has been implemented in ANSI C. The test sets have been created by SHA-1 hashing of consecutive numbers. The properties of SHA-1 guarantee that the resulting hashes are as random as they would be with using random input data. When an evaluation required different random sets, this was achieved by salting the hash. For the experimental evaluation we only considered synchronization in one direction. We created a set A of $1 \cdot 10^6$ elements. Because the hash trie overhead depends on the overlap of the sets, we created a set of test sets B containing

Algorithm 4 Trie traversal & comparison

Start recursion with WCBFS_Traverse($\{E_B$,*root*$\}$)

```
1:  procedure WCBFS_Traverse(E_B, to_check)
2:      dissimilar = CheckNodes(E_B, to_check)
3:      cand = {}
4:      for all node in dissimilar do
5:          if node.type = leaf then
6:              SendItems(E_B, node.hash)
7:          else
8:              cand = cand ∪ node.childs
9:          end if
10:     end for
11:
12:     while |cand| > collect do
13:         Push(stack, Pop(cand))
14:     end while
15:
16:     while |cand| < collect and |stack| > 0 do
17:         cand = cand ∪ { Pop(stack) }
18:     end while
19:
20:     if |cand| = 0 then
21:         return
22:     end if
23:
24:     WCBFS_Traverse(E_B, checkN)
25: end procedure
```

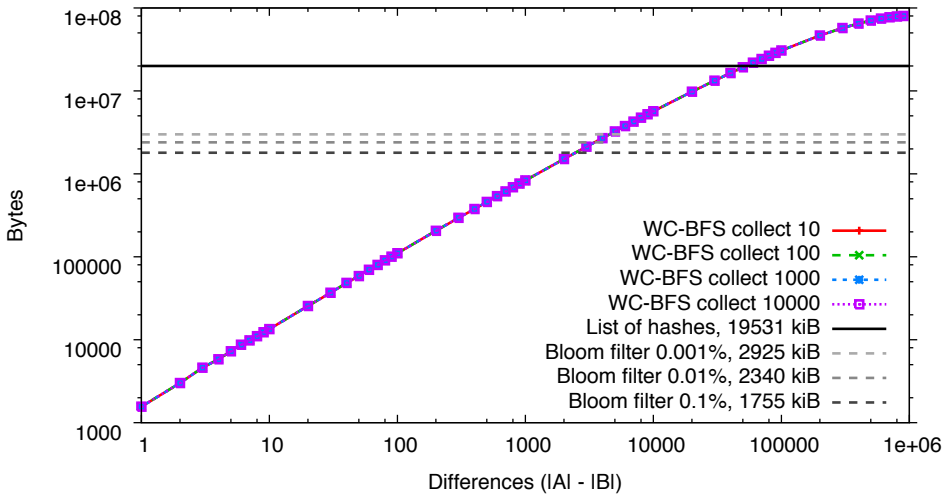


Figure 4.3.: Bytes transferred for $|A| = 1 \cdot 10^6$

different amounts of items from A.

We performed a synchronization between the complete $1 \cdot 10^6$ item set with each of the B sets. We measured the amount of bytes transferred and the total number of messages exchanged for each synchronization. The communication overhead only includes the amount of metadata including the hashes themselves that needs to be transferred between the entities and not the actual transmission of any user data. The amount of messages to compare a single trie node between two tries is 2: One message sending the hash and the node's position in the trie from E_A to E_B, and a second message containing the answer from E_B.

The amount of bytes to compare c nodes is $c \cdot 40 + \lceil c/8 \rceil$ bytes. We assume sending all c nodes we are interested in from the trie of E_A as a list. Each node in this list needs 40 bytes of transfer volume: 20 bytes are needed for the SHA-1 hash we want to compare, another 20 bytes are needed to send the prefix, to determine the position of that node in the trie. The prefix describes a path in the trie and thus uniquely identifies a subset of all elements in the set whose hash begins with the prefix. It will be used by E_B to traverse its trie to find the correct node to compare the hash value. The answer is a bit-vector of size c containing a 1 at position i, if the i'th node in the list has the same hash value in both tries, which signals two equal subsets.

Transferred Bytes

Figure 4.3 shows the amount of bytes transferred for a synchronization. The x-axis shows the number of elements missing in set B, e.g. when comparing the full set A with a 999000 item set B, the number of missing elements is 1000. The y-axis (log-scale) shows the amount of bytes transferred between E_A and E_B.

The "list of hashes" line shows the communication overhead of the naïve approach. This overhead is constant, as always all $1 \cdot 10^6$ hashes of set A need to be sent. The graph also

shows the overhead of a Bloom filter configured for a false positive rate of 0.1%, 0.01%, and 0.001% respectively. The Bloom filter overhead is independent from the overlap between the sets. Finding the best parameters for a Bloom filter can be tricky, and apart from theoretically optimal parameters also depends on the application, e.g., if the application limits the maximum size of a Bloom filter, or computational resources do not allow a large number of hash functions. For this experiment we set the maximum number of elements the Bloom filter should contain to $n = 1 \cdot 10^6$ and define the acceptable upper false positive rate p. Then the size m in bits of the filter is calculated as [74]:

$$m = \left\lceil \frac{n \cdot log(p)}{log\left(\frac{1}{2^{log(2)}}\right)} \right\rceil \tag{4.5}$$

and the number k of used hash functions is determined by

$$k = \left\lfloor \frac{log(2) \cdot m}{n} + 0.5 \right\rfloor \tag{4.6}$$

The overhead of the trie-based approach using different parameters for the *collect* threshold in the WC-BFS traversal is shown by the solid colored lines in Figure 4.3. The value of the *collect* threshold has almost no impact on the amount of bytes transferred, as instead of sending each node alone, several nodes are batched into a single message, which still needs the same amount of bytes to encode node data. In fact performance is a bit better for higher *collect* thresholds, as also answers will be batched, so it leads to less unused bits in the answer. As we assume byte level granularity of the communication channel, the answer for checking 1 node always "wastes" 7 bit.

As predicted, the overhead of the hash trie approach is prohibitive, if the sets differ too much. In the worst case it uses slightly over 4 times as many bytes compared to the naïve approach of transmitting just a hash list. This is due to the fact that instead of a 20 byte hash, for each node in the trie another 20 byte prefix needs to be transferred to describe the position of a node in the trie. This has to be done for $2n - 1$ nodes, which is the size of a binary tree with n items. Some additional overhead is caused by the feedback that needs to be transmitted from E_A to E_B. However, for small set differences the hash trie overhead becomes much more competitive, and drops well below the communication overhead for the approximate Bloom filter, which makes the trie-based approach a very good solution for situations where the difference between two sets is small: It uses less overhead, and does not suffer from the false positives introduced by a Bloom filter. For illustration Figure 4.4 takes a closer look (linear-scale) at the first part of Figure 4.3 for a collect threshold of 1000, when the sets are almost equal and the trie overhead is competitive. Different Bloom filter configurations are included for reference.

Exchanged Messages

Figure 4.5 shows the amount of exchanged messages for the same test. Here it can be seen that a higher *collect* threshold can drastically increase performance by reducing the

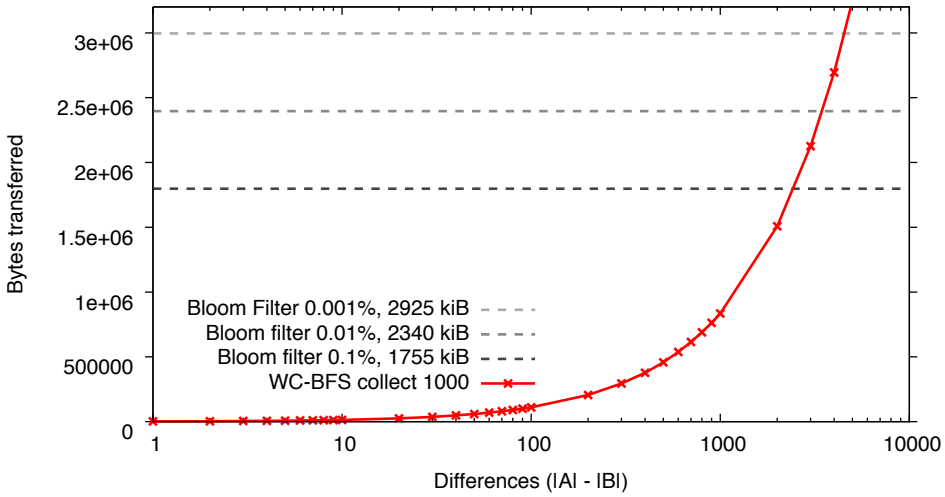


Figure 4.4.: Bytes transferred for almost equal sets ($|A| = 1 \cdot 10^6$)

necessary round trips. This is especially important if the hash trie synchronization is used in a network environment with potentially high latencies. For very small set differences there is no significant difference in the number of messages. This is due to the fact that in this case in the beginning of the synchronization enough subtries can be pruned, so the stack never grows large enough to fill the set of nodes to check in the next iteration up to the *collect* threshold. When the differences become larger, the stack gets populated, so that nodes can be batched. However, with parameters where the batching just begins to be effective, there is often the situation that only a single set of *collect* nodes can be sent from the stack, which is then almost empty again and the next round-trip is not able to send a complete batch. Therefore, the message overhead for these cases is worse, than for situations when the *collect* batch size can always be utilized fully. Later, when the stack is sufficiently populated, the differences between the different *collect* thresholds become obvious. In the extreme case, when there are $1 \cdot 10^6$ differences, every node from T_A needs to be transmitted and checked. As a trie with $1 \cdot 10^6$ entries consists of $2 \cdot 10^6 - 1$ nodes, roughly $(2 \cdot 2 \cdot 10^6 - 1)/collect$ messages are necessary. This is also confirmed by the graph, where for example the curve for $collect = 100$ tops out at ≈ 40000 messages.

The total amount of exchanged messages might seem high, and other set reconciliation approaches, which focus on using only a single round-trip such as a common Bloom filter variant or the method proposed in [80], exist. However, those single round-trip approaches are always approximate and for most practical applications the number of round-trips are not a problem: Firstly, since the WC-BFS can emphasize depth before breadth, the relevant metric are the number of round-trips until the first missing item is found: As in each step the algorithm will descend one level in the trie in case of a difference, the expected amount of round-trips depends on the trie's depth. The depth of a perfectly balanced binary trie with n elements is $log_2(n)$ for all paths from root to leaves. As our trie is expected to be reasonably balanced due to the randomness of the employed hash, we expect to find the first

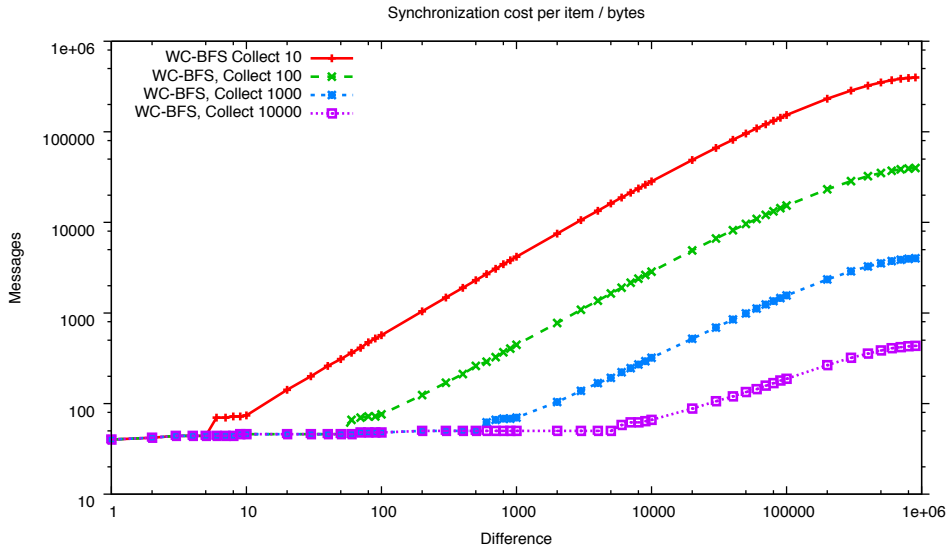


Figure 4.5.: Messages exchanged for $|A| = 1 \cdot 10^6$

difference after approximately $log_2(n)$ round-trips. Then, depending on the application, the first item needs to be transferred. Assuming that using the synchronization approach described here only makes sense when the data items themselves are reasonable large compared to the SHA-1 hashes, an application will probably use a protocol such as TCP that sends the item in segments and acknowledges their reception. In this case it is easy to piggy-back the remaining synchronization overhead on the goodput, which means no additional round-trips and latencies will be introduced.

4.6.2. Combined approach

When combining the Bloom filters with the hash trie-based synchronization mechanism the challenge is to find a trade-off between Bloom filter size and the expected overhead of the hash trie synchronization. This is depicted in Figure 4.6. It breaks the total communication overhead down to the amount used by the Bloom filter and the communication overhead from the hash trie-based approach.

We take the $1 \cdot 10^6$ item set from the evaluation and assume that B already contains 900000 items. If $|B|$ would be very small, the Bloom filters would yield a very low false positive rate and thus the overhead of the hash trie would be negligible. We start by choosing a maximum acceptable false positive rate for the Bloom filter. The parameters for the Bloom filter are calculated based on $n = 1 \cdot 10^6$ according to equations 4.5 and 4.6. Since the value for k is rounded, and is calculated for the complete $1 \cdot 10^6$ set and because the filter is later applied to a set with only 900000 elements, the resulting amount of false positives differs from the selected false positive rate. The gray bars in Figure 4.6 shows the size of the Bloom filter, and the expected number of false positives when using the calculated m and k parameters to construct B_B. As the false positives will be the differences between A and B, the red bar shows the communication of the trie-based synchronization for that

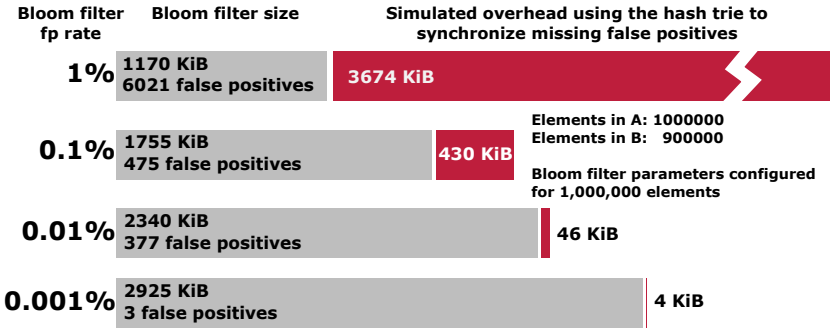
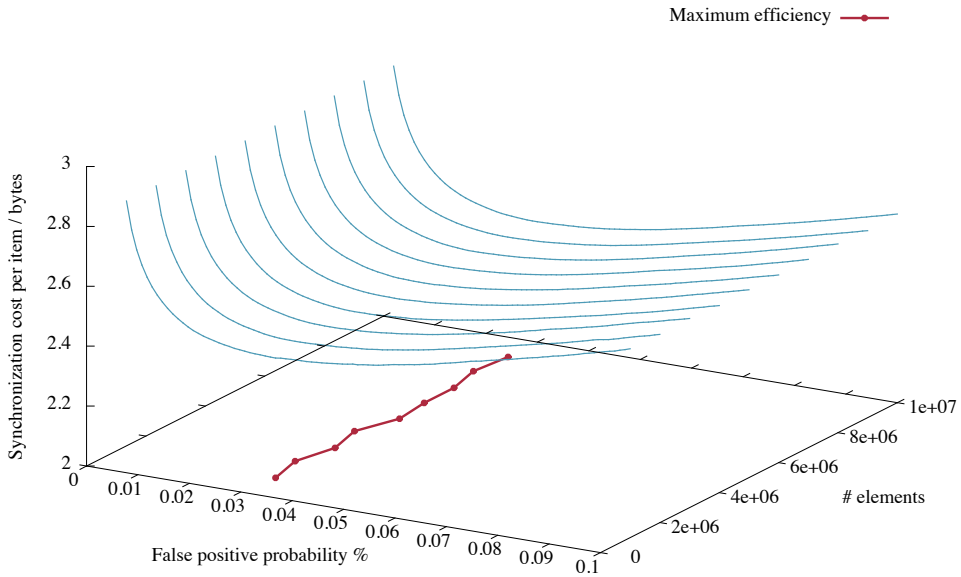


Figure 4.6.: Tradeoff between Bloom filter size and trie performance

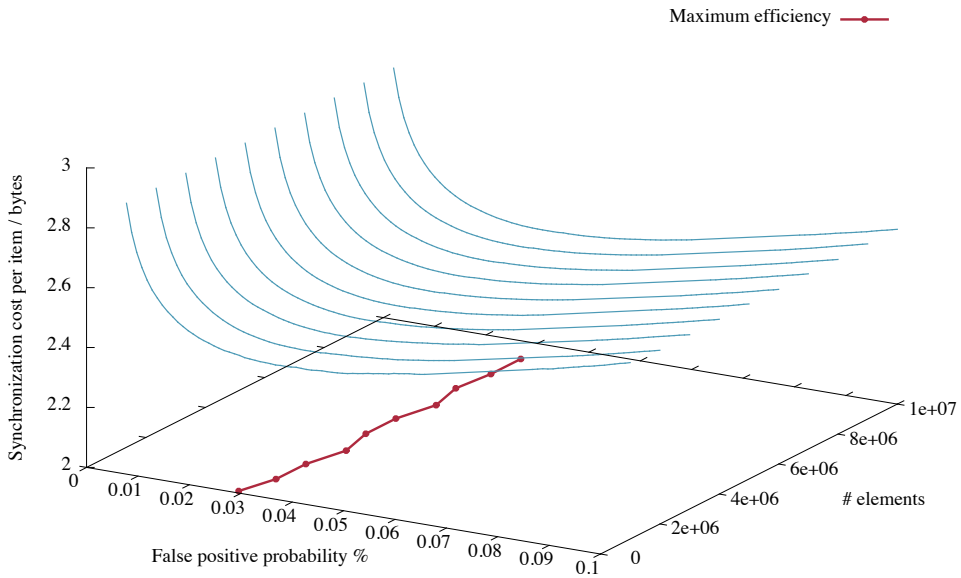
amount of differences using the trie-based synchronization. It can be seen that there is an optimal set for the Bloom filter parameters, which minimizes the total cost. If the Bloom filter is too small, there will be too many false positives and the hash trie performs badly. If the Bloom filter is too large, bandwidth is wasted transmitting it.

Looking at the examples in Figure 4.6, it can be seen that the optimum regarding the communication overhead is located between 1% and 0.01% false positive probability of the Bloom filter. To illuminate this further, Figures 4.7 and 4.7 show the influence of different parameters on the synchronization overhead. The x-axis varies the false positive rate, the y-axis varies the size of set B. A was chosen so that $|A| = x \cdot |B|$ with $x > 1$ and $B \subset A$. This was done to ensure, that B_B is sufficiently crowded so that false positives will occur, but to have a difference that is large enough that the false positives leave enough to do for the hash trie phase. The Bloom filter parameters were determined based on $n = |A|$. The z-axis shows the synchronization overhead normalized to the number of items checked ($|A|$). In all cases it can be seen that, just as expected, with a small false positive rate of the Bloom filter its overhead will grow exponentially due to the size of the Bloom filter. Similarly, when the false positive rate of the Bloom filter is too high, overhead grows, because the trie is inefficient due to many differences caused by false positives.

The red line in the plane spanned by $|B|$ and the false positive rate is the projection of the minimum communication cost for each $|B|$. In this setup for $x = 1.001$ (Figure 4.7a) the optimal false positive rate of the Bloom filter should be $\approx 0.03\%$ for all sizes $|B|$. The graphs show that the efficiency per item is better for situations with larger x (less overlap, Figures 4.7a-4.7d). In these cases the minimal cost is found for higher false positive probability rates: As the Bloom filter should always be configured for the maximum expected size of the complete collection (here $|A|$) it will obviously yield less false positives when checking against a smaller set using the same Bloom filter parameters. When the sets are very similar and both entities contain almost the complete collection, i.e. B_A and B_B are filled to their designed capacities, the false positive rate will reach the designed value. Due to the effect of a less crowded B_B for $|B| > |A|$ the per-item overhead will also decrease when applying the optimal false positive rate of 0.03% from Figure 4.7a to the scenarios with less overlap, i.e. for $x = 1.001$ the overhead per item checked is ≈ 2.4 *bytes* for $|A| = 5 \cdot 10^6$ while it decreases to $\approx 2.38,\ 2.26,\ 2.18$ for $x = 1.01,\ 1.1,\ 1,2$ respectively.

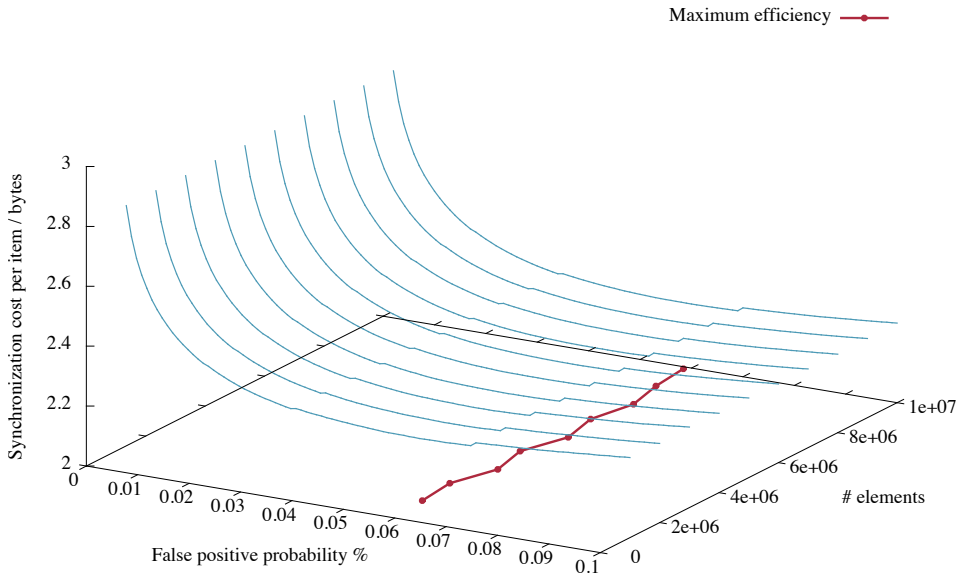


(a) Initial overlap: $|A| = 1.001 \cdot |B|$

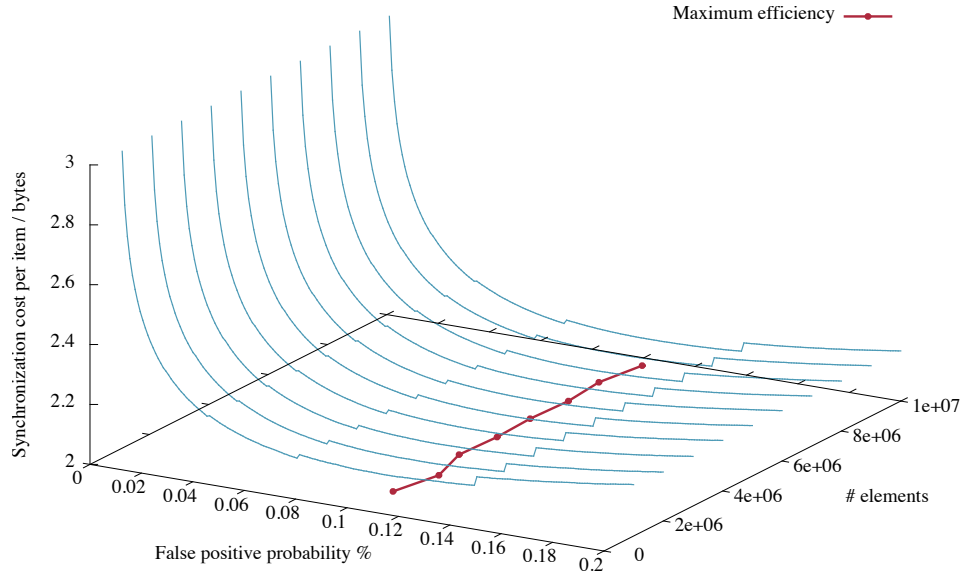


(b) Initial overlap: $|A| = 1.01 \cdot |B|$

Figure 4.7.: Bloom filter size trade-off, overhead per item



(c) Initial overlap: $|A| = 1.1 \cdot |B|$



(d) Initial overlap: $|A| = 1.2 \cdot |B|$

Figure 4.7.: Bloom filter size trade-off, overhead per item (cont.)

4.7. Theoretical Efficiency Analysis

4.7.1. Trie size

While the total amount of data that needs to be transferred when using the combined approach to synchronize two collections is small, both entities need to store the complete trie for their respective collections. As the tries are a full binary trees, the number of nodes in a trie representing a collection of n items is $2 \cdot n - 1$. In our implementation we use SHA-1 hashes, which are 160 bits long. For each node the implementation needs to store:

- The hash value (20 bytes)
- The prefix describing the position of the node in the trie (20 bytes)
- The prefix mask, describing how many bits to use from the prefix mask. 0 (root) to 160 (leaf) bits (1 byte)
- Pointers to the left and right child as well as to the parent (3 x 8 bytes on a 64 bit machine)

This gives a memory footprint of 65 bytes per node. Thus, a trie for $1 \cdot 10^6$ elements would consume roughly 62 MiB. This shows that for many use cases it is feasible to hold the data structure in RAM. In our implementation the performance of the data structure itself is mostly limited by the hashing function: In our profiling runs, hashing accounts for 80% of the total computation time. The synchronization itself will be limited by the speed of the network. Therefore, it should also be practically feasible to implement the trie as an on-disk data structure with no degradation of performance in practical applications.

4.7.2. Hash Trie Communication Overhead

In this section we present an analytical model, that can estimate the expected costs caused by the trie-based synchronization when the sizes of the two sets are known. Initially we will not consider the influence of the *collect* threshold in the WC-BFS traversal, i.e. the model assumes $collect = 1$.

We consider hashes of length m. This gives 2^m possible hash values. Let us further assume $|A| = i$ and $|B| = j$, with $i > j$. Consequently, the probability for a random hash value to be included in a set is $\frac{i}{2^m}$ for A and $\frac{j}{2^m}$ for B.

General Case for Random Sets

The trie-based approach compares subsets of the hash space. To answer the question whether we need to traverse a node, we need to find out if the subsets represented by this node are different or not. For the random case we assume that the elements in A and B are drawn i.i.d. from the set of all possible hashes.

The probability Pr_a, that k of the i elements chosen by A are contained in an x sized chunk of the hash space can be modeled by a hyper geometric distribution:

$$Pr_a(k,x,i) = \frac{\binom{i}{k}\binom{2^m-i}{x-k}}{\binom{2^m}{x}} \quad ,with\ k \leq i,x \tag{4.7}$$

where m is the length of the hashes in bit (160 for SHA-1).

Now the probability that two corresponding subsets of size x between A and B are similar can be given as

$$Pr_{sim}(x) = \sum_{k=0}^{min(i,j,x)} Pr_a(k,x,i) \cdot Pr_a(k,x,j) \cdot \frac{1}{\binom{x}{k}} \tag{4.8}$$

The probability of two subsets on trie level l being equal will decide the average fraction of costs that is to be expected for comparing all nodes in level $l+1$. So if the cost for traversing down a node is c, the expected total cost for synchronizing two sets can be given as

$$tc = \sum_{l=0}^{\lceil log_2(i) \rceil} \left(1 - Pr_{sim}\left(\frac{i}{2^{l-1}}\right)\right) \cdot 2^l \cdot c \tag{4.9}$$

It can be seen that in this general case the hash tries perform very badly, as Pr_{sim} will be very small in most cases due to Pr_a being small. Because the hash space is very big compared to the number of elements that are considered to be in the sets A and B, the high binomial coefficients make it numerically challenging to calculate Pr_a with an acceptable error margin. However, since for numbers of $i, j \ll 2^m$ $Pr_{sim} \approx 0$ the total expected cost can be estimated as

$$tc \approx \sum_{x=0}^{\lceil log_2(i) \rceil} 2^x \cdot c \tag{4.10}$$

that is, the algorithm needs to traverse all nodes.

Second Phase after Bloom Filter

If we apply the hash trie synchronization after two sets have been synchronized using Bloom filters, the situation looks different. We consider the use case from the evaluation: Synchronizing from a complete set to another partial subset: If A contains i items and B contains j items, with $i > j$, we know that $x \in B \rightarrow x \in A$. The missing items $\{x | x \in A$ **and** $x \notin B\}$ are those items in A, which gave a false positive when checking against B_B.

When traversing T_A the question is still, what the probability is that the subset $a \subseteq A$ represented by the current node is also in B. Therefore, the probability $Pr_{\Delta sim}$ that two corresponding subsets of size n between A and B are similar is now equivalent to the probability that a random drawing of n elements from A contains 0 elements from $A \setminus B$. This probability can be modeled as

$$Pr_{\Delta sim}(n) = \frac{\binom{i-j}{0}\binom{i-(i-j)}{n-0}}{\binom{i}{n}} = \frac{\binom{j}{n}}{\binom{i}{n}} \tag{4.11}$$

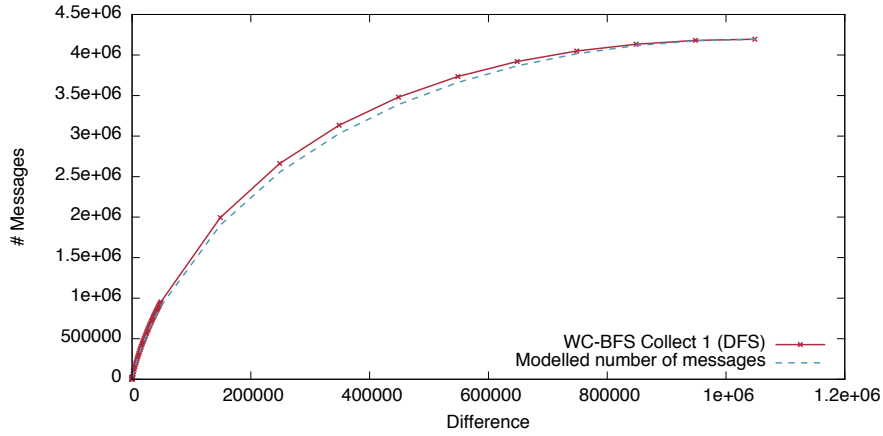


Figure 4.8.: Modeled and measured overhead for a set of 2^{20} elements

For a completely random hash function we can assume the trie to be balanced. Therefore, a node at level l will represent a subset of size $\frac{i}{2^l}$. This gives the following cost function:

$$tc = \sum_{l=0}^{\lceil log_2(i) \rceil} \left(1 - Pr_{\Delta sim}\left(\frac{i}{2^{l-1}}\right)\right) \cdot 2^l \cdot c \tag{4.12}$$

As this formula assumes a constant cost factor for each node it does not take the collect parameter of the WC-BFS into account, i.e. it calculates the overhead of a DFS (WC-BFS with collect=1) approach. When assessing the number of transferred bytes this is sufficiently accurate, however, for calculating the round-trips it is not. A reasonable estimate of the WC-BFS round-trips can be achieved by dividing the *tc* by the *collect* threshold as long as it is sufficiently small, and the tries are different enough, so that after the first few steps it can be assumed that the stack is full enough to always be able to send *collect* candidates.

Figure 4.8 shows the expected performance calculated by the model compared to the measured performance metrics. The model slightly underestimates the overhead, as the real tree will not be perfectly balanced, and thus needs to compare more and smaller subsets than expected by the model.

4.8. Comparison with Other Approaches

In this section we compare the presented approach to other set reconciliation methods. Difference Digests [83], Approximate Reconciliation Trees (ART) [81], Characteristic Polynomial Interpolation (CPISync) [82] and the extended Bloom filter approach (BloomTrie) presented in this chapter are considered. This comparison is based on the data presented in [83]. The test setup is as follows: There is one complete set of 100 000 elements and the second set is a subset of the first. The amount of data transferred during set reconciliation between two entities is measured in kiB. As a baseline, a list based approach is included. Figure 4.9 shows the reconciliation overhead of the different approaches. It should be noted that the naïve List, CPISync and the extended Bloom filter always find all the differences.

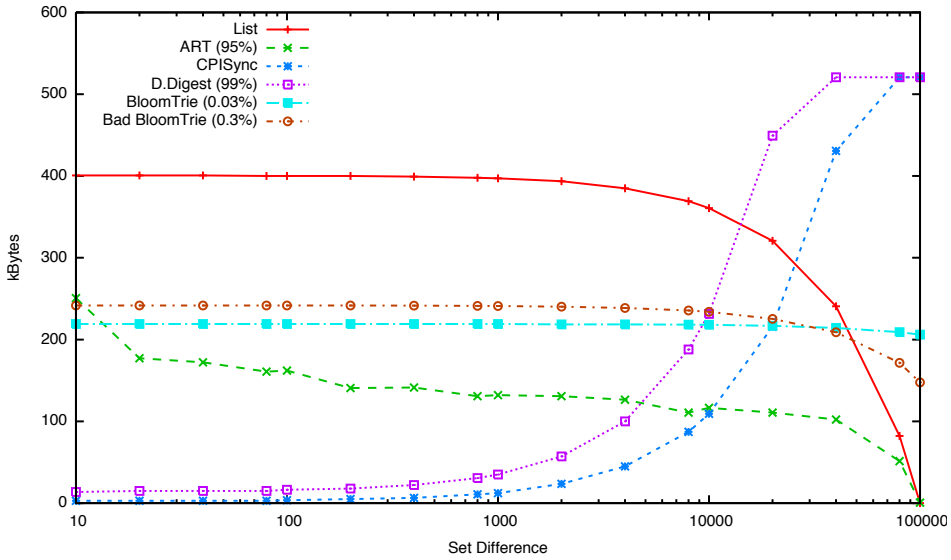


Figure 4.9.: Set reconciliation technique comparison

This is not true for ART: While the ART data structure is similar to the one presented here, the algorithm is not designed to retrieve the complete difference set as it focuses on using one round-trip only. For the measurements shown in Figure 4.9, ART has been configured to retrieve 95% of all differences 99% of the time.

While Difference Digests will eventually find all the differences, there is a chance that the set reconciliation will fail with the IBF size chosen by the difference estimation. Here, the Difference Digest is parametrized in such a way that it will be able to recover the differences in 99% of the time [83]. In 1% of cases the IBF can not be decoded completely, which will add some additional overhead for transmitting a larger IBF in a recovery phase (not shown). For maximum efficiency CPISync needs to know the difference set size beforehand. The figure shows the best-case overhead for CPISync.

In this comparison the BloomTrie needs more overhead for large and small set differences than some of the other approaches. However, the BloomTrie shows the most consistent behavior: The overhead is fixed regardless of the set difference, as it is dominated by the Bloom filter itself, while the trie does not add any significant overhead. Similar to plain Bloom filters it is enough to know the expected size of the complete collection to parametrize the BloomTrie algorithm. According to the results in Section 4.6.2, the Bloom filter should be configured with 0.03% false positive probability. However, the BloomTrie approach reacts gracefully to non-optimal parameters. If an application severely underestimates the collection size and uses a Bloom filter size which yields a false positive rate that is 1 order of magnitude larger than the optimal one (0.3%), Figure 4.9 shows, that the overhead for the BloomTrie is still very reasonable. For smaller set differences the additional overhead is around 20 kiB. The reason is that the too small Bloom filter generates more false positives, which need to be fixed by the more costly trie approach. These costs will overcompensate the savings due to the smaller Bloom filter. For larger

	CPU	L2-Cache	RAM	Operating System
PC	Intel i7 950 3.07 GHz (4 Cores+HT)	8192 KiB	12 GiB	Ubuntu 12.04 LTS 64 bit
VM	AMD Athlon II X4 630 2.81 GHz (1 Core)	512 KiB	512 MiB	Ubuntu 12.04 LTS 64 bit
Laptop	Intel T2500 2.0 GHz (2 Cores)	2048 KiB	3 GiB	Ubuntu 12.04 LTS 32 bit

Table 4.2.: Evaluation hardware

set differences the mis-configured Bloom filter actually saves some overhead. In this case the Bloom filter is more sparsely populated for the smaller set, so that it still yields an acceptable false positive rate.

This stability makes the extended Bloom filter a good choice for practical applications: Estimating the size of the complete set is much easier than estimating the size of the difference set which is necessary for approaches such as Difference Digests and CPISync. Trivially, for two sets A and B the size of $A \cup B$ will be $max(|A|, |B|) \leq |A \cup B| \leq |A| + |B|$, which is good enough to parametrize the Bloom filter. A practical heuristic for choosing the Bloom filter would be taking the upper bound of the set size and using the compressed Bloom filter variant[82]. This would yield optimal performance for large sets, while still saving space for a smaller set when the filter is only sparsely populated.

4.9. Practical Performance

While the combined approach theoretically has a good performance due low network overhead, the question is whether there are practical performance limits for implementations. Two operations are crucial for the synchronization approach presented here: Hashing and adding and retrieving hashes from some sort of persistent database. As items are represented as numbers, hashing might be used to generate unique numbers from arbitrary data. This is a problem when building a collection for the first time, or when adding elements. But hashing is also constantly needed during operation: Keep in mind that the trie demands constant rehashing of path from leaf to root, whenever an element is added or removed. Because the synchronization finds out identifiers of missing elements, a reverse lookup table from hashes to actual elements is needed. Especially for large selections the lookup should be fast enough.

4.9.1. Hashing Performance

To get an idea about hashing performance we tested hashing performance on three different machines listed in table 4.2. Please note that the system “VM” was a virtual machine running on a 4-core AMD host. Only one CPU has been assigned to the VM. This CPU has 4 MiB of cache, with 512 KiB assigned to each core. The laptop uses a first generation Intel “Core”

Input Data	System	OpenSSL	PolarSSL
40 Bytes (2*SHA-1)	PC	503.112 ms	477.156 ms
40 Bytes (2*SHA-1)	Laptop	1 622.320 ms	1 701.320 ms
40 Bytes (2*SHA-1)	VM	675.278 ms	607.652 ms
1024 Bytes	PC	2 523.980 ms	3 491.180 ms
1024 Bytes	Laptop	7 442.060 ms	11 502.100 ms
1024 Bytes	VM	3 504.710 ms	4 503.680 ms

Table 4.3.: OpenSSL vs. PolarSSL: Time for 1 million hashes

processor which is not 64-bit capable.

We used the SHA-1 hash function we recommended for the trie. We tested two implementations: The OpenSSL crypto library[4], as this is one of the most-widespread security libraries that is available for most operating systems. As an alternative we used the SHA-1 implementation from the PolarSSL library[5]. Compared to OpenSSL, PolarSSL is a smaller, more modular crypto library. For small embedded targets, when the full OpenSSL implementation is not an option, the PolarSSL implementation of SHA-1 can easily be pulled from the sources and added to another application without further dependencies.

To simulate hashing up to the tree root, we used both implementations to hash 1 million pairs SHA-1 values (40 bytes input for each hash). To see whether the performance difference changes for larger input data (i.e. when hashing user data) we also performed 1 million hashes of 1024 byte input data. The results of this test are shown in table 4.3. The faster implementation is highlighted for each of the test-cases.

For the small test case representative of rehashing the trie PolarSSL wins, except on the old 32 bit laptop. However overall, the results are close together. Requiring between 0.5 and 1.6 seconds for one million hashes should not pose a problem for a real world application when syncing two sets. For larger input as in the 1024 byte test-case OpenSSL takes the lead. This indicates that the OpenSSL implementation has higher initial setup costs, but higher throughput. In both cases the older 32 bit laptop exhibits significantly lower performance than the newer platforms. Whether hashing of application data will be a problem, depends largely on the size of the data that needs to be hashed and thus on the application. This is independent from the hash trie and would also be needed for vanilla Bloom filters or for rolling checksums such as used by rsync.

Especially for common hashes such as SHA-1 there is an increasing number of platforms supporting hardware acceleration for this operation. VIA introduced the “PadLock Hash Engine” with the second generation PadLock hardware acceleration modules to its low-power x86 processors in 2005 [88] which claims to have a peak throughput of 5 GBits for SHA-1 and SHA-256. Often crypto accelerators are found in Embedded SoCs to offloading these computation heavy operations from the slower CPUs. An example is Marvell’s

[4] `http://www.openssl.org/`
[5] `https://polarssl.org/`

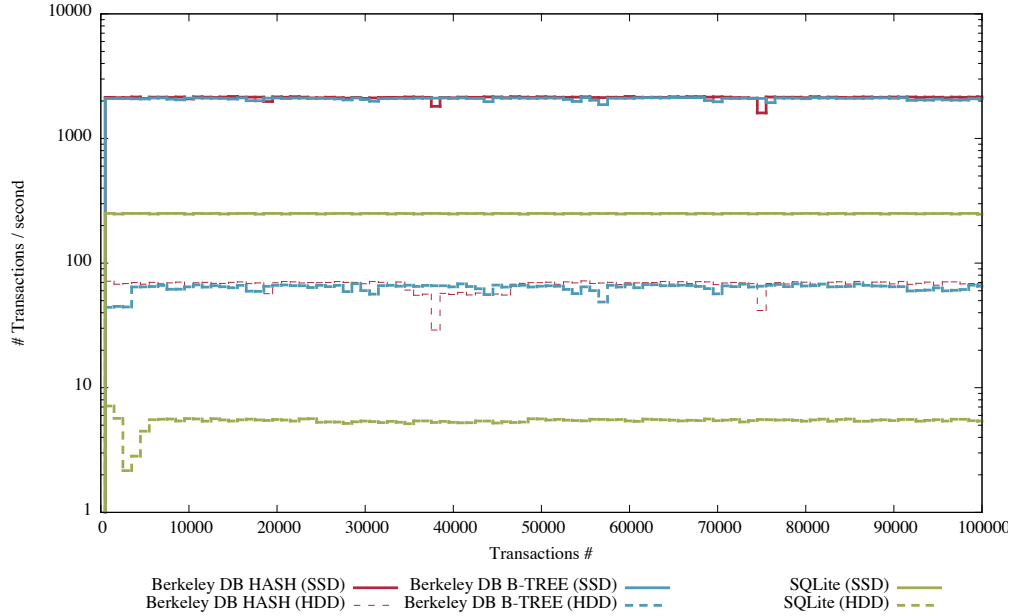


Figure 4.10.: SQLite vs. BerkeleyDB

"Cryptographic Engines and Security Accelerator" (CESA) [89] that can be found in many of the company's ARM cores. Apart from several encryption algorithms CESA supports SHA-1 and MD5 hashes. Broadcom offers a large family of MIPS SoCs for networking equipment such as SOHO routers that also offer a crypto accelerator capable of SHA-1 [90].

Overall hashing speed is not a problem today, and especially embedded processors generally found in small network appliances such as routers or NAS devices, often support hardware acceleration.

4.9.2. Hashing Caches

For the Bloom filter and the trie many hashes need to be stored and retrieved. We tested what kind of real-life performance can be expected. Generally a key-value store is needed. We compared SQLite[6] with BerkeleyDB[7]. SQLite is a complete relational SQL database, and thus provides more than just the required key-value functionality. It is already integrated into a lot of software (IBR-DTN also includes it), and thus might be a sensible choice for storing hashes. BerkeleyDB supports key-value stores. We performed 100 000 transactions adding hash values to the databases. BerkeleyDB offers 2 options for data organization: B-tree or Hashtable. We tested both configurations. The solid lines in Figure 4.10 show the transactions per second on an SSD-based system database, while the dashed lines show the performance on a HDD.

In all configurations, whether running on an SSD or running on a HDD, BerkeleyDB significantly outperforms SQLite. We also performed the test on a VM that was hosted on

[6] `http://sqlite.org`
[7] `http://www.oracle.com/us/products/database/berkeley-db/overview/index.html`

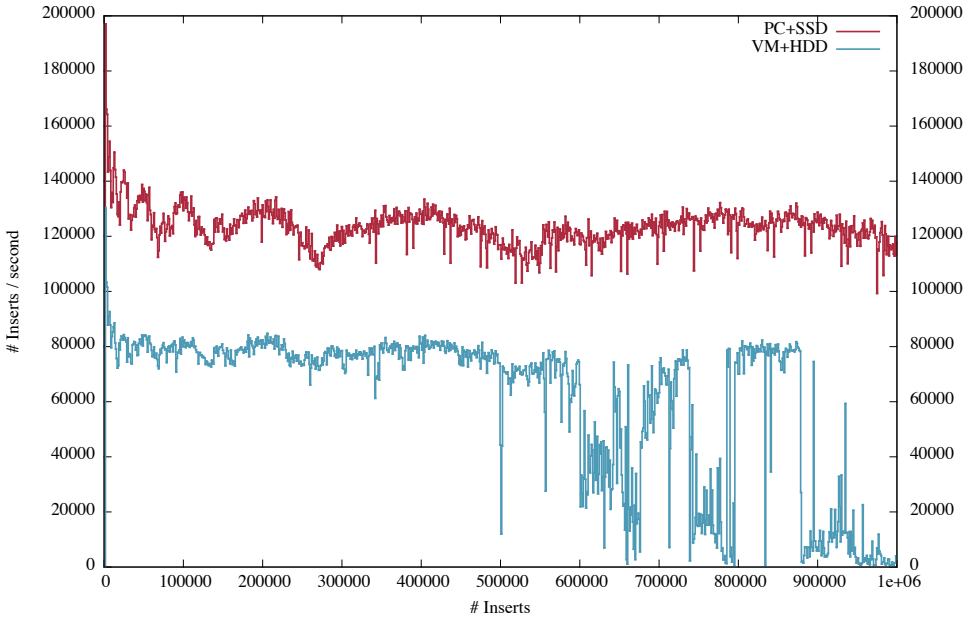


Figure 4.11.: Adding $1 \cdot 10^6$ elements (k=20 bytes, va=200 bytes) to BerkeleyDB

a HDD and restricted to 512 MiB of RAM (not plotted). The other systems where equipped with 12 GiB of RAM. As we have used the transaction system on both databases, RAM should not have an influence as data will be regularly written to disk. Surprisingly, the VM outperformed the HDD setup. We assume the comparatively good values for the VM were due to the fact that the host system performed some caching not seen by the guest.

When adding larger amounts of data, the speed of the disk becomes relevant. For the test in Figure 4.11 we added 1 million key-value pairs to BerkeleyDB. We used a 20 byte key (SHA-1) and a 200 byte value, that represents a link to real data such as a file. While the SSD system can sustain around 120 000 insertions per second, the low-memory HDD-based VM system begins to struggle after 500 000 insertions. Whether this is a problem in real systems, depends on the expected use-case. In many systems dropping huge amounts of new items into the storage at once should not happen, as the arrival rate of new elements is limited by the available network capacity.

During synchronization mostly the database needs to be queried to find data associated with a specific hash. Figure 4.12 shows the performance that can be expected when continuously querying for hashes. As can be seen, the database will not be a bottleneck for the synchronization approach presented here.

The remaining question is, how does the on-disk size of the database scale. It is to be expected that a database occurs some overhead compared to an in-memory data structure. To test this we added 10 million entries to a BerkeleyDB with hash-table organization. Every 1000 entries the database was flushed to disk and the resulting file size has been plotted in Figure 4.13.

It can be seen that BerkeleyDB extends the underlying file in intervals. Overall the

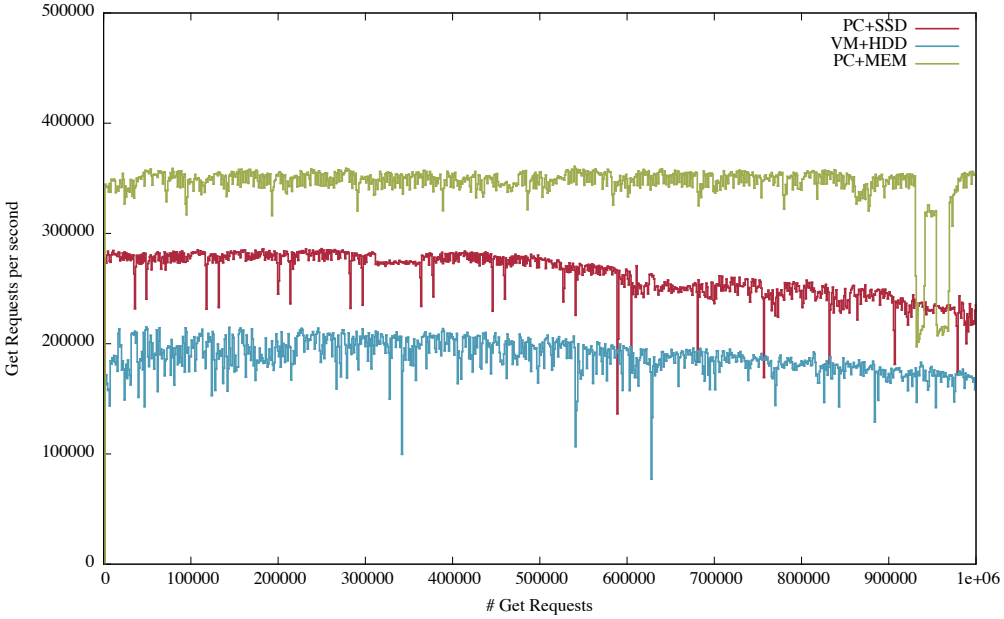


Figure 4.12.: Querying BerkeleyDB with $1 \cdot 10^6$ entries

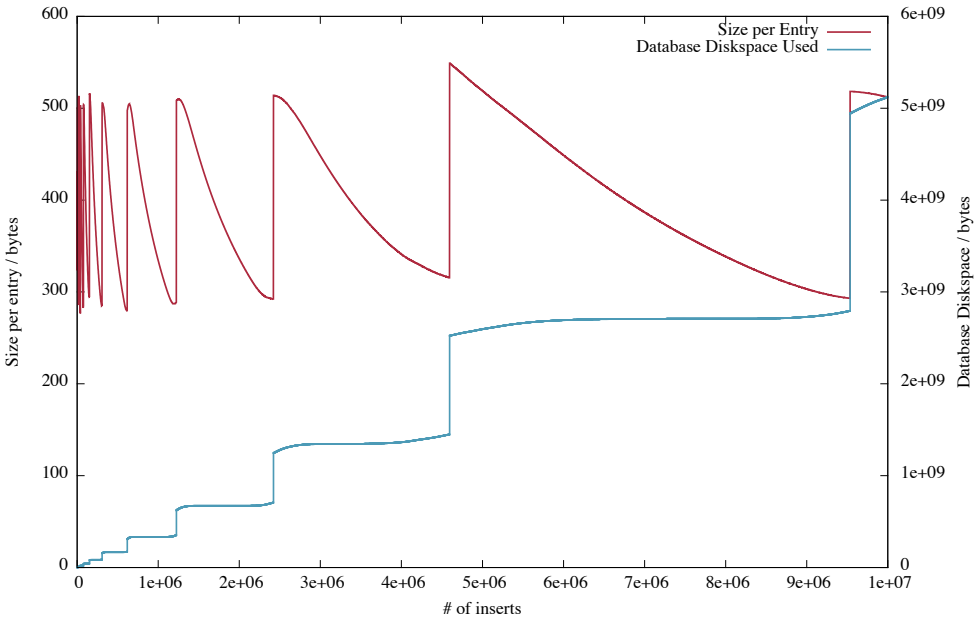


Figure 4.13.: File size when adding $10 \cdot 10^6$ entries (k=20 bytes, va=200 bytes)

	Distance	Propagation Delay	Roundtrip
Theoretical Minimum	$54.6 \cdot 10^6 km$	182 s	364 s
Theoretical Maximum	$401.0 \cdot 10^6 km$	1342 s	2684 s
Average	$205.0 \cdot 10^6 km$	751 s	1502 s

Table 4.4.: Distance and propagation delay between Earth and Mars

storage efficiency is quite good with roughly 60 byte overhead per entry when the allocated space is used almost completely before the next extension phase. As with many databases, deleting entries does not reduce the size of the database file. However, the freed space in the file will be used for subsequent additions to the database. The BerkeleyDB API also supports compacting the database, but this operation is slow, and should only be used for maintenance.

4.10. Large Propagation Delays

We have shown that the synchronization approach presented here is a good fit for stateless synchronization between two DTN nodes. It is stable, performing well over a wide range of parameters and for practical purposes as efficient as a pure Bloom filter approach but without the false positives. For shorter contact durations the *collect* can be used to fine-tune the search strategy between DFS and BFS to enable the reconciliation to find distinct elements faster. Thus, the parameters such as the Bloom filter and the *collect* parameter can be optimized to get the best contact utilization for a given scenario. This makes the presented approach a good fit for opportunistic networking scenarios with a large number of bundles: Whenever nodes come into contact, no matter their history, they can use the contact efficiently to reconcile their bundle storages.

However, in systems with large propagation delays such as in interplanetary scenarios the round trip delay is the limiting factor. As an example lets consider a communication link between Earth and Mars. In recent history Mars has been closest to earth in 2003 at $56 \cdot 10^6 km$. As Mars and Earth have different orbits, there are many possible configurations. Table 4.4 shows the theoretical minimal and maximal distance. This gives an idea about the round trip times that can be expected ranging from about 5 minutes to 25 minutes. Keep in mind that this implies line of sight. In many configurations, when the sun is between Earth and Mars communication would be blocked leading to larger delays.

The propagation delay in itself is not a problem. Where efficiency is lost is during query response phases: In the Earth/Mars example up to 25 minutes of a communication window are lost when one side needs to wait for a response from the other communication partner. In the approach presented in this chapter this happens on several occasions: First, the Bloom filter is sent. Then an entity waits until it receives the missing items determined from the Bloom filter phase. Much more round-trips are necessary in the trie phase: Whenever up to *collect* nodes are sent from the trie, a node waits for the response before continuing.

In this section we will suggest some extensions to the BloomTrie approach to enable streaming operation. We will assume a full duplex channel and aim to utilize the bandwidth in both directions with useful data. To achieve this, we first look at different kinds of information exchanged between two entities during synchronization:

- Bloom Filter: Bit array containing the Bloom filter for the first stage.
- Data Item: When the Bloom filter or trie phase identifies a missing element on the other node it will be sent. Once a data item is received, it can be integrated into an entities Bloom filter and its trie by using its hash. For the streaming approach we make a clear distinction between the hash and the actual associated data. If an entity receives only the hashes it can already update its trie and Bloom filter to represent the state as if the object has already been received.
- Trie Node: One or more trie nodes are sent during the trie phase. This allows the receiving entity to check whether it already possesses the subset (or single item) represented by the node.

We assume that it is preferable to use a link in a not space-efficient manner instead of not at all. This is true, when the synchronization is the only traffic on a link. Usually it is preferable that the gaps due to round-trips can be used by other applications using th same link. However, especially for interplanetary links it is common that there is only a finite communication window. In this case it is more important finishing the synchronization task in time instead of optimizing bandwidth usage.

Figure 4.14 shows how the synchronization time can be utilized more efficiently under large propagation delays. One entity begins by starting to transmit its Bloom filter. Even in this scenario depending on the application it is not advisable that both nodes transmit their Bloom filters at the same time: While that would lead to 100 % bandwidth utilization it will prevent actual transmission of user data until the Bloom filters are completely transmitted. Furthermore, as we have shown in Algorithm 3 it might be more efficient to just request the missing hashes directly, after receiving the other Bloom filter. An important fact about the Bloom filter is that the receiving side can begin to process it, even before it fully arrived. As soon as a 0 position is detected, where the receiving entity has some item hashing to that location, it can begin to transmit that data while continuing to receive the remainder of the Bloom filter.

There is an important difference when using the BloomTrie synchronization over links with long propagation delays: When operating in this mode an entity will always transmit hash information before transmitting the actual item. When there is no time constraint for synchronization hashes do not need to be explicitly transmitted at all: An entity can just recalculate them after having received an item. The advantage of transmitting hashing information is that the receiving node can immediately update its trie structure to a state indicating the item has already been received. This allows the trie phase to commence before the actual items identified by the Bloom filter phase have been transferred completely.

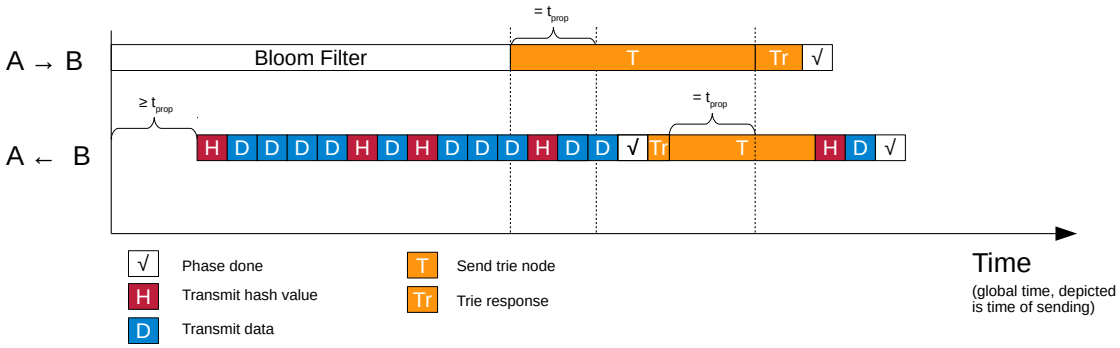


Figure 4.14.: Streaming set reconciliation for large propagation delays

After the Bloom filter has been transferred, the entity will begin transferring trie nodes. Since it will take at least 2 times the propagation time until the other entity can react, the sending entity can choose to start with a much lower trie level and transmitting more nodes at once. In the example in Figure 4.14 it takes even longer, as there are still some items from the Bloom filter to be send, after the first trie request is received. After answering trie requests, a node can already send nodes from its own trie instead of idling. An optimized algorithm should aim for sending different nodes than the ones that it expects to receive next. A simple algorithm would be to bias traversal to left nodes on one entity and to right nodes on the other. Due to this interleaving it might happen, that entity A sends a trie node before receiving missing elements from the Bloom filter that would change that node. This is however not a problem, as after the missing hash has been incorporated into the trie, the checking of the next lower level of the trie would catch subsets that now became similar due to the new item. Again, space efficiency has been traded for better usage of available time.

4.11. Summary

In this chapter we looked at the problem of synchronizing bundle storages without any prior context. This is a challenging problem in a DTN if contact times are limited and the number of active bundles is high. If possible, the most efficient way is synchronization with context. This however is only possible, when the routing is completely deterministic or single-copy routing schemes are used. And even in those cases there can be failure modes, such as a crashed DTN node, that demand recovery of the synchronization state without any reliable context information. In opportunistic scenarios, with replicating routing schemes, where it is unknown what the history of a contacted node is, context-less synchronization is a requirement.

As exchanging lists of complete collections does not scale well with larger sets, the Bloom filter provides an efficient and well proven method for set reconciliation under these conditions. However, the false positives make it an imperfect choice for a DTN. Therefore, we suggest extending the Bloom filter with a second trie-based phase based on full SHA-1 hashes for removing the false positives. This improves the reliability of the reconciliation

to the same level as using a list comparison, without increasing the overhead significantly compared to a simple Bloom filter. Compared to other set reconciliation approaches the Bloom Trie approach is computationally easy and very robust with regard to the input parameters. Practically speaking, the hashing and the management of the Bloom filter and trie are the most important factors determining the performance of the approach. As we have shown, hashing is not a problem for modern systems and often hardware acceleration is available. As any DTN node managing large amounts of bundles already needs a powerful storage subsystem, managing the Bloom filter and trie data-structure on-disk should not pose a problem as well.

5 Incentives for Users

5.1. Problem Statement

In the previous chapters we have discussed how to scale DTN technologies to the Internet. Today a significant chunk of the Internet is represented by mobile devices such as smartphones. In the last years the smartphone has become an integral component of many research projects focusing on ubiquitous computing. It is often proposed to exchange data relevant to the application between individual mobile phones. Such networks became known as Pocket Switched Networks (PSNs) [26]. A more recent trend is public sensing: Here we are interested in the sensory information that a smartphone can measure or collect and which needs to be transported to some central sink. These use-cases are premier examples for the application of DTN technologies. In most of the proposed systems, not much thought has been given to the question *why* an individual user should provide bandwidth and energy to transport data for the application at hand. This, however, is a crucial consideration for enabling mobile smartphone-based DTNs: Why should users take part in such networks, when they need to invest energy, communication and storage capabilities? Do users demand sufficient immediate benefit from whatever application he is using, or can they be convinced to provide a general service to the network, without knowing what services specifically profit?

In this chapter we will design an incentive system which is especially tailored for smartphone users in urban DTN networks. We will introduce the stakeholders in such a mobile phone based DTN system, and show how to realize and secure such a system. We will estimate the investment and operational costs to show that such a system would be economically feasible. In the second part of this chapter, we present a prototype system which has been tested by users. The goal was to find out whether users are able to operate the system, and if the offered reward was deemed acceptable. We have been using an advanced questionnaire, to get an idea what kind of rewards would work, and more importantly whether it is possible to sustain users' motivation to participate in such a network.

Parts of the work presented in this chapter have already been discussed in [91, 92].

5.2. Related Work

Ever since the idea of ubiquitous computing has been introduced [93] and continued to evolve into the recent Internet-Of-Things meme, it has been clear that one important basic concern for any distributed system is the sharing of resources, such as computational power, storage capacity or network bandwidth. An important aspect of the current situation is that many of those ubiquitous devices such as smartphones, tablets, game consoles, NAS

devices and of course also the old-fashioned PC, are *personal* devices: They are paid for and used by individuals. Thus, whenever the idea of sharing or donating resources comes up, immediately one question needs to be answered: Why should an individual give away a share of his resources. What is the benefit?

This question has only been sort of solved for two applications: P2P file sharing and volunteer computing. In file sharing, where the goal is to download a certain resource, variants of the tit-for-tat strategies have been proven to be very successful: To download a certain item you are interested in, you are required to upload an appropriate amount of material interesting to others [94]. For scientific computations the so-called volunteer computing is a viable alternative to buying or renting big compute clusters. Private users can install a small program on their PCs to help out a research project with some spare CPU cycles. The main motivation for such science projects, are lower costs compared to classic compute clouds [95], while the motivation for participants is comparable to other charities: A feeling of doing the right thing, supported by a sense of competition as all projects publish detailed statistics which participants contribute most. The aforementioned examples of resource sharing work because they focus on powerful PC platforms: For a user there is no or little impact when participating in these schemes.

An interesting target for sharing communication capabilities are smartphones. Many approaches have been proposed using them to create an ad-hoc network independent of cellular infrastructure. The vision is that everybody has a smartphone in his pocket that opportunistically connects with other devices in range and exchanges data [26, 96]. Usually, these kinds of networks are DTNs which means when there is no suitable communication partner in range, a device will store network data until the next communication opportunity. These networks are a good basis for participatory sensing [97, 98] which becomes increasingly popular within the context of smart city technologies. Other approaches suggest using smartphone-based networks to offload data from a congested cellular network [99, 100]. To encourage equal participation in such systems, there has been research into fair routing protocols using tit-for-tat strategies [101, 102]. The systems proposed in [103, 104] show how to securely built an incentive system on top of a DTN. However, in both systems the incentive itself is not clearly defined, but rather an abstract token.

The problem with these ideas when applied to smartphone-based networks is that sharing resources of a smartphone definitely affects battery life. The user experience could be negatively impacted. On the other hand, the benefits for users in the aforementioned applications are more insubstantial: *Somehow* a smart city can work better due to the data collected by a phone or *somehow* the network performance increases. But these are not immediate benefits, such as getting a desired file from a P2P network.

We think that, despite a solid and secure technical implementation of an incentive system, the question *what* a feasible incentive is, has not yet deserved enough attention, yet it is the most important question when designing an application: *Why* should users support my system? Can I pay them enough, or offer sufficient benefits to motivate participation? These incentives need to be economically feasible. Giving more expensive phones to users

for free and even paying them as proposed in [100] probably does not fall into the area of "economically feasible".

5.3. A Game-based Incentive System

In this Section we propose a game-based incentive system that combines gaming and advertising to build an economically self-sufficient DTN network that motivates private users to participate. Basically, the proposed system is an advertisement and entertainment platform that is able to support data proliferation in a network.

The proposed system

- allows any entity, regardless of size, to participate as a service provider at low cost,
- allows for joint campaigns between local advertisers without any middle-man,
- will actually be *fun* to use by smartphone users,
- works by implementing proven technologies from the area of DTN,
- is able to improve data proliferation for other services such as public sensing applications in a DTN network.

5.3.1. Location-based Services

Looking at the development of the Internet in general, we see in the past 10 to 15 years the so-called social networks led to the partial transcendence of our daily social interactions from the physical world [105]. During this time our ability to interact in these web communities got more ubiquitous: Instead of having a PC as the single portal to these virtual communities, today we can carry the window to our virtual peer group with us at all times in the form of mobile devices such as smartphones, tablets or smartwatches. Mobile ubiquitous Internet access brought back the idea of localized services: As mobile devices have a good idea of their current position thanks to GPS and Wi-Fi/Cell tower-based triangulation techniques, this is used in a number of applications: "Bump" applications, such as the original Bump[1] or competitors such as Hoccer[2] transfer data between two gadgets when users bump their devices together or perform certain movements. This is done by correlating the data from accelerometers (the "bump") with time and position of the devices. Another famous example for Location Based services are "Check-In" system such as Facebook Places[3] or Google Latitude[4]: Based on their location users can virtually "check in" into certain physical locations such as shops, public places or even private homes. This information can then be posted on a social network site and shared with other persons within the same network. Recently, Facebook started offering a special service on Facebook Places in order to entice more users to use it: When checking into the location

[1] http://bu.mp/
[2] http://hoccer.com/
[3] http://www.facebook.com/places/
[4] http://www.google.com/intl/en_us/latitude/

of a cooperating business, a user can redeem a virtual coupon that might provide some discount at the place or entitle the user to some free merchandise. FourSqare[5] is offering a similar service.

A system for distributing personalized advertisements to smartphones has been suggested in [106]. However, the focus in [106] is to collect user profiles using a DTN while protecting users' privacy. For the dissemination of the ads themselves the authors suggest using making use of specified, but so far largely undeployed, Multimedia Broadcast Multicast Service (MBMS) capabilities for 3G networks.

While these services offer the illusion of being executed in a local context, in fact they all operate through a central entity operating the service. This limits availability and reliability of such a service. Centrally operated services can also make it difficult for other businesses to join them and profit from the infrastructure. Some services may be closed by design, while for others it is a business model to sell access to these services. Even though today many of these offers are still operating in "startup mode", it seems to be only a matter of time until the dominant operators with the highest market penetration will charge their customers for promotional campaigns.

5.3.2. A DTN-based System

Functionalities such as offered by the aforementioned systems can be replicated easily with DTN technology. Instead of mobile phones communicating their GPS positions to a central server, which is then able to simulate locality in the service, it is possible to really put DTN nodes at interesting locations relevant to an application. Thusly the system becomes really decentralized, since an open protocol is all that is needed to let everybody participate in such a system. While from an application point of view the same services that today are built on a central infrastructure can be realized, the advantage from a network point of view is that users interacting with such services can also act as data-mules transporting bundles from location to location. Furthermore, due to the locality, applications might provide a better quality of service. In the following sections we will introduce a game-based system designed to encourage people to visit various locations with their smartphone. This system is designed to be self-sufficient when it is used for advertisement purposes, while at the same time there is the additional benefit that the network capacity generated by users can be used for all kinds of smart city applications.

We assume that users have Wi-Fi capable smartphones with a special app. This app will connect users to so-called Smart Ads. A Smart Ad is basically an access point whose existence is known or detected by the application. When connected to an access point, bundles might be exchanged. More importantly from the users' perspective access to a Smart Ad might trigger a reward for the user such as a discount coupon. We designed the system in such a way that *explicit* interaction with the system is necessary: The application will not be scan for access points or other devices in the background. As we have discussed in Section 3.2 all mechanisms for the discovery of opportunistic contacts consume significant

[5] `https://foursquare.com/`

energy. Therefore, the system only scans for and contact access points when the application is brought to the foreground by the user. This keeps users in full control of their devices. As the challenge posed by the game will encourage users to go to specific access points to exchange data, we assume that such an approach will result in more usable contacts at much lower energy consumption compared to opportunistic systems running in the background. Of course, once a battery technology improves and energy-efficient scanning is available, the system could easily be extended to support a fully opportunistic mode running in the background. However, the developments in the last few years do not give rise to the hope that a breakthrough in any of these fields is near.

5.3.3. Smart Ad Games

In this section we give an overview of five different games a Smart Ad system might support. A game usually takes the form of a simple quest that a user can be asked to fulfill. If applicable, for each type of game we give a rationale, how it could be used by advertisers, and how it could be used by entities interested in the data carrying abilities of a user's device. Of course in real deployments both types of stakeholders can be mixed.

Freebies

Every user connecting to a Smart Ad gets a coupon entitling him to some sort of bonus or discount at the provider. This is similar to coupons which today are mass-mailed or even commonly offered for download on web pages of big franchises such as Subway or Burger King. An example for a GUI presentation for this basic kind of coupon can be seen in Figure 5.1a. This is a very basic form of advertisement, enticing customers to do business with the advertiser. In case many advertisement campaigns run at the same time, an individual user might visit several Smart Ads, improving data proliferation in the network in general, however data will not be directed.

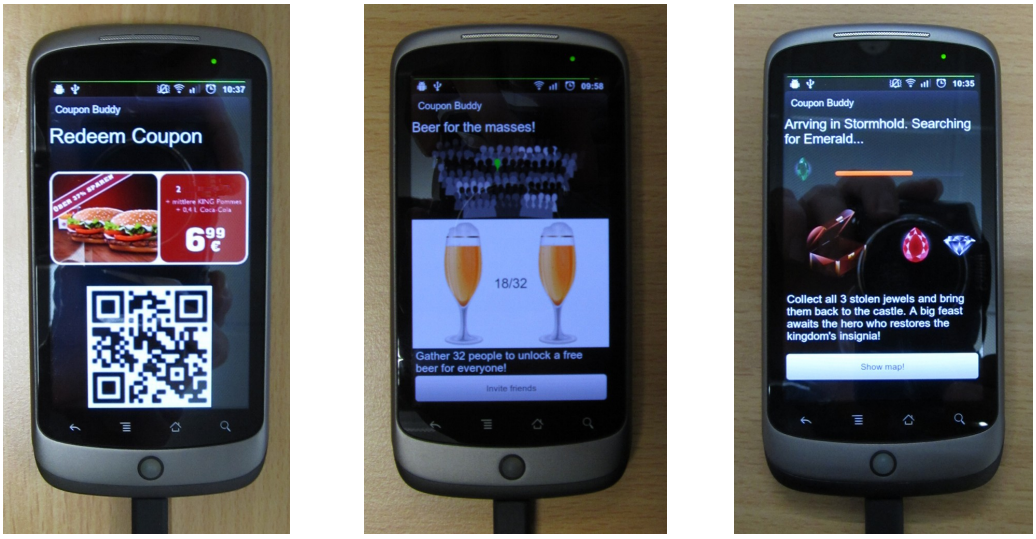
Lottery Game

To receive a reward in this game the task is to find a certain item. The user has to connect to a Smart Ad in order to "search" for the item. For pure commercial scenarios, the item to be searched should be located at one of the provider's Smart Ads. For scenarios which aim at optimizing data flow for third party applications the "search" time at each Smart Ad is used to transfer data to and from the device. At every Smart Ad there is a certain base-probability to find an item. This probability is increased depending on:

- the amount of data a user delivers to a Smart Ad and the amount data he has already propagated in the network while searching for the item
- the relative need of the Smart Ad to receive data, i.e. a badly connected Smart Ad is more likely to give prices

Gathering Crowds

If a certain amount of people logs into a specific Smart Ad during a specified time frame, each is eligible for a reward. For businesses this can be used to lure customers into the



(a) Redeeming a coupon (b) Collecting game (c) Crowd game

Figure 5.1.: GUI prototype

venue, for data proliferation this can attract a crowd of mules if there is a big amount of data that the station needs to distribute. An example for a GUI visualization can be seen in Figure 5.1b.

Matching Game

A user starts with an item received from a Smart Ad and his goal is to find a matching item. Either the user can be directly told where to go, or he is given a number of locations which he needs to search. This can also be combined with the lottery game (see Section 5.3.3).

For advertisement applications the user can be sent to another cooperating provider: The pizzeria's Smart Ad says: "Go to the flower shop now, buy your friend a nice rose and receive a coupon for a free drink with your next pizza". Only Smart Ad users who first connected to the pizzeria's Smart Ad will be offered the respective coupon at the flower merchant. If the goal is to support third party applications' data flow, the system can use this game if it wants to transfer data to a specific destination.

Collecting Game

A user has to collect n several items from different Smart Ads before receiving his reward. For advertisers this can be a means to raise their visibility: "You have three days to visit *all* our locations. We offer every successful combatant a free dessert with your next dinner and you will have the chance to win the special price!" If the goal is to support third party applications' data flow, this is a good incentive to increase general data propagation in the network, e.g. proliferating broadcast information. An example for a GUI visualization can be seen in Figure 5.1c.

5.3.4. Security Issues and Architecture

It is to be expected that users will try to exploit the proposed system. We will discuss the requirements for security and look at possible attack vectors in order to derive a suitable security architecture. To ensure data integrity we assume that each Smart Ad has a private key that it can use to sign any data it sends out.

Identity

Usually security becomes much easier, if each user is known to the system and can be identified and authenticated if needed. However, this would be very hard to realize in the completely distributed system described here. Moreover, making users register first would be counter productive as the goal is to motivate as many users as possible to participate in the system. A unique ID might still be desired, so that certain phones can be identified, when interacting with different kinds of Smart Ads for some of the presented games. Therefore, we propose just using a randomly chosen UUID as identifier for an individual user.

Double Spending

Double spending is the problem of redeeming a coupon more than once. Whereas paper-based vouchers can be discarded in the shop, this is not advisable for smartphones. However, the implications of double spending are not as severe as it seems: Many current coupon distribution schemes have the same problem. Especially food franchisers regularly allow customers to print their own coupons by downloading them from the respective franchiser's website. In principle every person can print as many coupons as he likes. This is already included in the calculation and usually a coupon includes terms and conditions only allowing one to be used per purchase.

The Smart Ad solution even allows a tighter security than paper-based schemes: The Smart Ad appliance can number each coupon before sending it to the smartphone and only distribute a fixed amount of coupons for each campaign. This number can be encrypted with the Smart Ads private key and put into a QR code as seen in Figure 5.1a. Upon scanning in the shop, the number can be decrypted and a database can be consulted to make sure each coupon is spent only once. For even tighter security requirements, identification schemes outside of the Smart Ad system can be used: For high value promotional actions, such as a subsidized mobile phone there is the need to make an explicit contract between customer and seller. In this case, a "one-benefit-per-customer" rule might be enforced by allowing the advertised offer only once for each household.

Sybil Attacks

If the identity of users is only an easy to forge UUID as postulated in Section 5.3.4, it is easy for a user with a modified software client to pose as many users, and thus being able to gather much more coupons from a single Smart Ad appliance. Here basically the same recommendations as in the double spending case apply: Either it is not so important, or you need to incorporate unique customer identity later in the process.

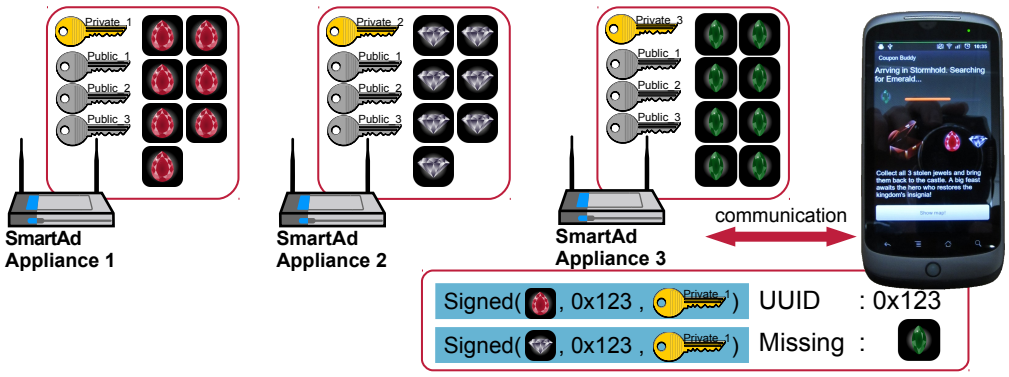


Figure 5.2.: Smart Ad cooperation security model

Copying of Coupons

Another attack would include copying legit coupons from one smartphone to another. This will hurt the system, especially if those coupons would be distributed via the Internet, which precludes users from having to visit the locations physically. For simple promotional campaigns this problem is negligible if it does not occur on a large scale, as users still have to visit a venue to redeem a coupon, which was the goal after all.

This problem is more severe if the original goal was to increase data proliferation in a DTN, as users who do not visit Smart Ads cannot transfer any data. However, this behavior can be discouraged by introducing a unique number into each coupon that can only be spent once as outlined in Section 5.3.4.

Security of Joint Campaigns

For games requiring visiting more than one Smart Ad we will use the following abstraction for the security model: Every game involves that the user collects *tokens* from different Smart Ads. Depending on the game, there might not be a required order for visiting the Smart Ads, and the decision whether a given user gets a token from a specific Smart Ad might be probabilistic. Once a user has all the required tokens, he is issued the final coupon.

When a quest demands visiting several Smart Ads, the system must make sure that each Smart Ad can verify what the user already did, i.e. whether he concluded all the necessary steps to be eligible for a token from this station. For this we propose the following system: For a campaign each participating Smart Ad appliance possesses a public/private key pair. The public keys of all Smart Ads participating in a campaign need to be known to each other. During the setup phase the keys might be distributed through some Internet backend or, for small setups, it might be feasible to generate the campaign on one appliance and put the configuration data on a flash drive, that can be used for importing the necessary keys and descriptions into the other participating appliances.

When a smartphone is in range of a Smart Ad appliance, it can offer its current tokens. If the requirements for a given campaign are met, the Smart Ad will give another token. A token consists of the user's id and the "part" of the quest the user completed, e.g. the items

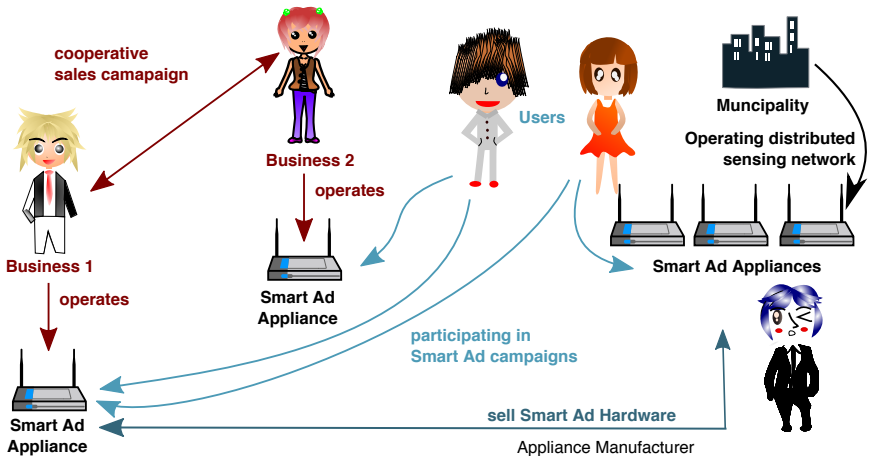


Figure 5.3.: System stakeholders

already found in a collection game. The token will be signed using the issuing Smart Ads private key. Depending on the requirements additional encryption could be provided, for example using the public key of the next Smart Ad, if the game demands a sequential order. The distribution of keys and tokens is shown in Figure 5.2 for the situation depicted in the GUI prototype shown in Figure 5.1c. Here the user already collected the red and the white emerald from Smart Ad 1 and 2 respectively in a "Collecting" game. It is requesting the missing green emerald from Smart Ad 3. Since during the setup phase Smart Ad 3 got the public keys for this campaign form Smart Ad 1 and 2, it can verify that the user already visited the two locations required for the red and white emerald and thus it will provide and sign a token that verifies the user gained the last green emerald. In case the user fulfilled all requirements of the game, Smart Ad 3 will also issue the coupon.

5.4. Economic Feasibility

Figure 5.3 shows a system overview with all the relevant stakeholders. First there are the manufacturers of mobile phones and Smart Ad hardware (see Section 5.4.1), who can sell devices to users of the system. As a Smart Ad will run on existing hardware platforms, vendors just gain an additional customer base; there is no need to develop anything new on the hardware side. However, there is of course the possibility to generate some extra revenue by independent vendors selling devices installed and pre-configured with a Smart Ad software stack.

User will like the chance to use their already existing smartphones to receive vouchers. For businesses this means they only need to invest in the Smart Ad hardware once, which will amortize itself quickly compared to printed vouchers, as the cost of operation apart from electricity are negligible. Being able to set up joint campaigns with other businesses is another bonus, which is not easy to realize currently. If a shop is already using a marketing agency for advertising campaigns, those agencies will be able to provide Smart Ad campaigns as an additional service at a low cost to their customers.

A public sensing network or other Smart City applications operated by a municipality might wish to use the Smart Ad platform to facilitate data propagation. The reward for participating users can also be vouchers from cooperating local businesses. As most municipalities have funds set aside for means to stimulate local economy, it can be argued to take the investments for a Smart Ad assisted data propagation network from those founds. This will still be cheaper than providing infrastructure to every corner of a city, especially for places which are slightly remote or where high bandwidth is needed.

5.4.1. SmartAd Hardware Costs

As shown in [15] IBR-DTN can run on a variety of low-cost hardware platforms. For a Smart Ad appliance a wireless router platform seems to be a good choice. A typical example for customizable wireless router platform would be the Mikrotik[6] RB751U, which has a suggested retail price of $59,95. The RB751U includes 802.11b/g/n WiFi has 64MiB flash, 32 MiB RAM, and a 400 MHz MIPS CPU. This is enough to run the embedded Linux distribution OpenWRT[7] with IBR-DTN and a small web interface to administrate the Smart Ad and set up campaigns. Add a USB thumb drive for storage ($\sim$ $5), and a Smart Ad can be built for considerably less than $100 including taxes. There are lots of other hardware possibilities in the sub $100 range. For the prototype system used in the user study (see Section 5.5) we used a system based on the Raspberry PI which cost around EUR 60. In a nutshell, the cost and hardware for a Smart Ad appliance should be very similar to that of typical SOHO routers commonly used today.

5.4.2. Cost Compared to Newspaper Campaign

We performed a cost analysis of developing the Smart Ad system. We assume that the Smart Ad system will be developed by a software company which already has experience in developing mobile applications. Further, we assume that only an Android version is developed, and that it will use IBR-DTN as a communication framework instead of developing a custom solution. We expect that the software can be developed within 6 months. The numbers are based on the business model of selling the software and service for 2 year aiming for a 10% profit before tax. If the company is operating in Germany this leads to total costs of $\approx$ EUR 190,000. Due to different tax regulations and income levels this number will vary for other countries. A breakup of the costs can be seen in Figure 5.1.

We assume a workload of two full-time software developers, a designer and a 50% workload for a project manager for 6 months. This leads to costs of EUR 80,000. After development the workload for supporting the running systems gets much lower. For the projected development we assume a 50% developer equaling to another EUR 40,000. Additional costs factors are soft- and hardware, rent and hosting costs. For selling software a turnover tax of 19% applies (EUR 31,475). We aim for a 10% profit (before tax) adding another EUR 15,600. This can of course only be a rough estimation. Spinning up a new company just for developing the Smart Ad system would incur higher costs, while for example an established

[6] https://www.mikrotik.com/

[7] http://OpenWRT.org/

Smart Ad Development	Cost/EUR
Developer Costs	80,000
Support Staff	40,000
Server/Hosting	400
Hard- and Software	10,000
Office Rent	6,000
Training	5,000
Profit	15,600
Tax Burden	31,475
Total	188,475

Newspaper Campaign	Cost/EUR
Advertising Cost	307,200
Tax Burden	21,504
Total	328,704

Table 5.1.: SmartAd development costs v.s newspaper campaign

ISV with some free capacities might save money due to low marginal costs.

Operation costs in the first year for a franchise with 80 subsidiaries including hardware investments (EUR 100 per subsidiary, see Section 5.4.1), electricity costs (assuming 10 watts consumption) and a back-end server would total about EUR 10,000.

As a comparison we consider a traditional newspaper campaign: Based on prices of the "Frankfurter Allgemeine Sonntagszeitung" (FAS, a nationwide weekly newspaper in Germany), the cost of adding 50 g of brochures is EUR 160 per 1000 exemplars excluding tax. In this case in Germany a turnover tax of 7% applies. Considering the FAS has a print run of 480,000 the total cost for the ad campaign is about EUR 330,000. This is only the distribution cost, and does not include the actual production of the brochures.

Of course, a real business has to consider their target market, as probably the Smart Ad system and the newspaper campaign will reach different demographics. Also, we did not include an estimation of the costs for rolling out the Smart Ad system, as these are highly dependent on the organizational structure of a company and hard to predict generally. In any case, the low operating costs of a Smart Ad system are hard to beat.

5.4.3. Spotawin

In the third quarter of 2014 the company Triology [8] actually tried to launch "*Spotawin*", which implements a very similar, albeit much simpler, concept to the Smart Ad idea presented here. Users are required to install the *spotawin* application on their Android or iOS device. The app will use GPS to spot nearby sweepstakes, which usually are on the premises of participating businesses. A user needs to walk to the indicated location and scan a QR code. This entitles him to take part in a prize draw.

The business model of *spotawin* is similar to the Smart Ad concept presented above: Companies offering the challenges are required to pay. The advertised prices asked by Triology are rather bold (see Figure 5.4). Customers need to pay a small fee (EUR 35.00) for

[8] http://www.triology.de

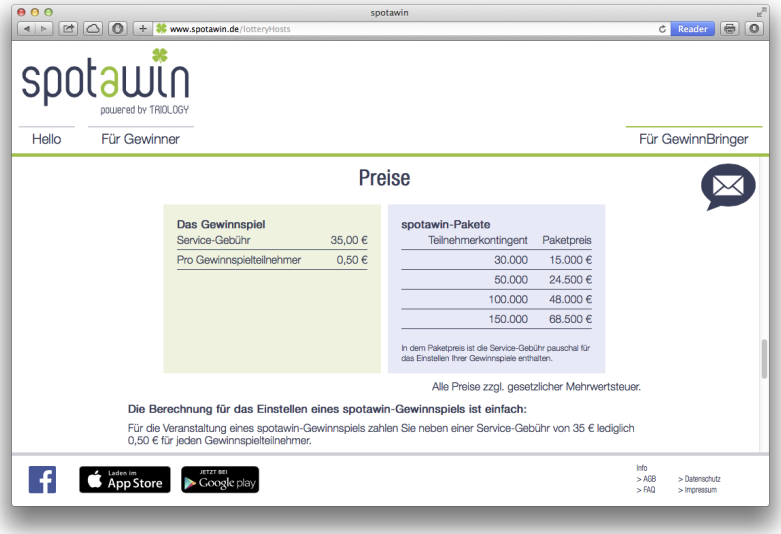


Figure 5.4.: Spotawin prices as of Sept. 17th, 2014

setting up a new location and an additional EUR 0.50 for *every* user that scans the QR code. If theses prices turn out to be enforceable in the market, building an ad-supported DTN would be commercially much easier than anticipated in the previous sections.

5.5. User Study

So far we have proposed a way to generate participation for realizing a DTN with reasonable network capacity. We demonstrated that the investments for such a system are rather low and proposed local business holders as stakeholders to provide the investment. The resulting network would be a good basis for Smart City applications. We assumed that the proposed games would be enough incentive to motivate users to take part in such a system.

The goal of this section is to check whether this assumption holds. While it is certainly possible to give a rebate coupon to a user for transporting some data halfway across the city, would people actually do it? We performed a user study to find out whether these rather symbolic rewards would be accepted by users. We need to answer the question, whether people can be motivated, and more importantly, whether the motivation is *sustainable*: Will people only use the system once, due to the novelty factor, or is it valid to assume that they participate continuously.

Generally speaking, asking people about their inner motivations or feelings is hard, and does not always yield the desired results [107]. To this end we decided to perform an actual user study: Implementing the user experience of the proposed system as close as possible to a commercially developed system, and let people use it. The user study deployment required participants to download an application to their mobile phone and reach several separate destinations within a city. After having experienced the system, a carefully crafted

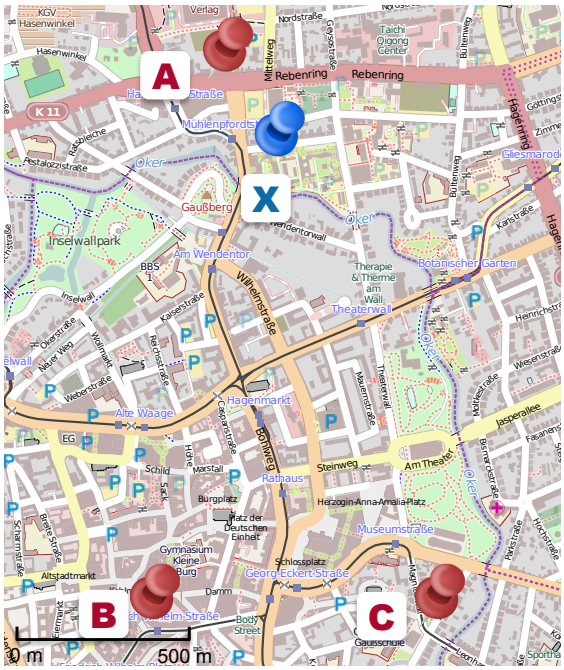


Figure 5.5.: Goals in Braunschweig

questionnaire and interview was used, to learn about the participants' experience and motivating factors.

5.5.1. The Game

According to the Smart Ad concept presented we implemented the user study as a game. The intention of the game was to send players to different areas with their smartphones or tablets. When enough users can be persuaded to move to specific locations a Smart City application could exploit the users' mobility and communication abilities to transport data between specific locations. It is important to note, that we did not communicate this potential use case to our participants: When starting the game for the first time, the user is informed that he has to reach 3 points, and that upon finishing that task he can fetch a reward consisting of sweets.

For this user study we used only one static set of goals: A participant is required to visit 3 destinations, distributed within an area of less than 2 km^2 in the city of Braunschweig, in order to successfully finish the game. The Braunschweig goals are marked with "A" to "C" in Figure 5.5. In a DTN, asking users to visit specific places might be done because there are some sensing stations or sinks belonging a Smart City sensing system. An application that requires users to transport data could be an electronic billboard system that needs to be updated with new advertisement videos.

After completion of all goals, a participant is asked to come to an office at the university (location"X") to collect his reward. Additionally, after reaching all 3 goals in Braunschweig and revealing the reward collection location "X", the game offers people to double their score by finishing an optional bonus goal. In order to see how far people would go, and

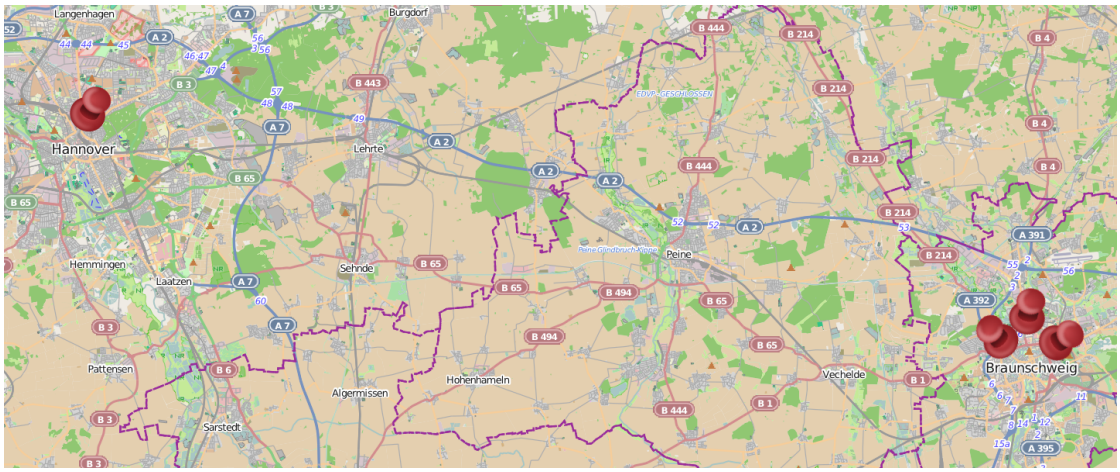


Figure 5.6.: Hannover bonus goal. 60 km distance to main goals

to make sure the effort cannot rationally be justified by the reward the bonus goal was located in another city, Hannover, about 60 km away from Braunschweig (see Figure 5.6). The bonus goal was located in a business area (see Figure 5.7d) in walking distance from Hannover main train station.

All 3 Braunschweig goals are known when a user starts the game, so it is possible to finish them in any order. We placed the goals in such a way that they are easy to reach. Point "A" (Figure 5.7a) is located directly besides the university main campus near the city center. Point "B" (Figure 5.7b) is inside the city center and point "C" (Magni District, Figure 5.7c) is located in a historic district near the center. Except the 1 month playing period of the experiment there is no time limit. We made sure the reward could be gathered anytime during working days.

We decided to make all interaction with the system explicit in our implementation. Many projects proposing smartphone-based networks assume that the necessary software will run in the background all the time, scanning for possible contacts and autonomously exchanging data. While this is a conceptually nice idea, as we have shown in Section 3.2, technology is not quite there (yet): Operating in such a way has tremendous impact on battery life as there is no technology which allows continuous energy efficient scanning for neighbors. Beacons need to be send and received. Both operations are expensive with any RF technology.

With the incentive system implemented for this user study this is not a problem, as the concept adapted for this study is designed to involve the user by offering challenges to him. For this use case an application running silently in the background would be rather counter-productive. Instead, the design demands users' to consciously move to the required places. Once near a goal, the user has to open the application to complete that goal. Only when the user opens the application and uses it as a foreground application it will acquire the current location and scan for a base station. Energy will only be used if the user actively uses the application. This makes the impact on battery life more predictable

(a) A: BS, near university

(b) B: BS city center

(c) C: BS, Magni district

(d) Hannover bonus goal

Figure 5.7.: Goals in Braunschweig (BS) and Hannover (H). Arrows mark the locations of the Wi-Fi access points

and keeps the control over a device completely in the users' hands. Of course, this design precludes opportunistic device-to-device contacts but one can argue that this does not matter much, as a smartphone-based network is inherently a DTN network. Therefore, the added delay due to potentially unused device-to-device contacts should not be a problem for applications running on such a network. Also, this effect is offset by the fact that with such a game-based approach participants can actually be *directed* to some degree, which improves network performance.

5.5.2. Implementation Details

The application was called "GeoGame" and has been implemented on Android. We put considerable effort into making the application as streamlined and intuitive as possible, in order to provide a user experience that is as close as possible to a fully functional deployed commercial system. Careful design and testing made sure that the UI provides good usability on the smallest mobile phone screens as well as on full size tablets (see Figure 5.8). The main part of the screen is the map, which shows the user's position and possible next targets. When starting the game for the first time, or when completing a goal, the user is informed by a message that will suggest possible next steps. Once a user has finished all goals and comes to the reward collection point, a slide show will lead him to the exact office within the university building.

We also implemented the Smart Ad base station. While we could just have used the location abilities of a smartphone to simulate the effect of reaching a DTN router, doing so would have altered the user experience. Our base stations provide Wi-Fi access points. Just as in a real deployment, the mobile application needs to detect whether it is in the vicinity of a base station, and then associate to the appropriate Wi-Fi network and begin exchanging data. This is the same procedure that would happen in a real DTN system and makes the user study more realistic in terms of reliability and latency when connecting to the base stations. Tests have shown that the whole process of connecting to base station and exchanging some information through a TLS connection can take up to 20 seconds.

We used RaspberryPi Model B SBCs[9] (Pi) as base stations. A Pi is a low powered ARM system running a standard Linux operating system. A Wi-Fi interface was attached via USB. The whole system could be powered by a standard micro USB mobile phone power adapter. This hardware setup costs significantly less than EUR 100 and thus, meets the cost efficiency of the solution proposed in Section 5.4.1. While in a real scenario the Pis could also represent DTN sources with no Internet connection, for the user study we connected all Pis to the Internet to make monitoring and maintenance of the experiment easier. The Pis have been deployed in residential homes or offices. A typical installation can be seen in Figure 5.9.

The Pis connect to a central database server. Whenever a mobile phone connected with a Pi, this was reported to the back-end. This information could later be used to check whether a person collecting the reward was really eligible. In a deployed DTN application

[9] `http://www.raspberrypi.org`

Figure 5.8.: Android GUI on a 10" tablet and a 3.8" phone

Figure 5.9.: Pi base station at goal A (near university)

a cryptographic token would be given to devices as proof of reaching a certain goal (see Section 5.3.4). In the user study the centralized approach made managing the system much easier, without altering the user experience. Only anonymous Android IDs have been collected in order to identify participants. The back-end includes a web interface which could be used to trace activity in the game. No personal data have been collected during the experiment. Due to the nature of the game, we needed access to the devices' location stack and a unique identifier for each device, to track the game's progress. The device identifier can not be connected to a particular person and any data generated during the game has never been transmitted to us, except the device id when a participant connected to a base station. When downloading the application from the Google Play Store, the provided description informed participants in detail which permissions the application requires and what those permissions have been used for. After conducting the study, all collected data from the application have been deleted.

5.5.3. Questionnaire

All participants who completed the final goal and came to collect their reward, had been asked to fill a PC-based questionnaire. During the process of filling the questionnaire a participant will receive his actual reward (see 5.5.5). When possible, questions are multiple choice, or use Likert scales [108] when asking questions of degree (e.g. "How important was the reward? – Very important / important / not very important / not important"). The questionnaire was conducted in German and consists of 8 parts that aim to highlight different aspects of the system.

- *General Questions*: General information such as age, gender or professional position are gathered in the first part. This demographic data can help to put other collected data into perspective.
- *Smartphone Usage and Familiarity*: We collected information how often participants carry their smartphones with them and whether they already had knowledge about or interest in other location based applications.
- *GeoGame App Experience*: We asked, whether there have been any specific technical problems or usability issues with the GeoGame application. Implementation-specific problems are unrelated to the type of system we wanted to evaluate and could bias the results.
- *Difficulty*: We measured the perceived difficulty of reaching the goals in the game. This part also includes questions about the mode of transportation used to reach a goal and the familiarity of the participant with the game area.
- *Optional Bonus Goal*: This part deals specifically with the bonus goal in Hannover. We asked for the reasons why a participant chose to complete the bonus goal or not. Data about the mode of transportation to reach the bonus goal has also been collected.
- *Trust*: As collecting potentially personal information is a sensitive topic, especially in Germany, we added a block to the questionnaire that assesses whether participants were convinced that the application did not collect personal information. We wanted to see, whether - or for what reasons - users trust the application or not.
- *Reward*: Before starting this part of the questionnaire, a participant gets his reward in the form of sweets. He could take as much as he wanted (see Section 5.5.5). The amount in grams is put into the questionnaire. We asked, whether the user considers the reward an important aspect of the game or not. We also checked whether participants are happy with their reward or whether they would have preferred something else.
- *Intrinsic and Extrinsic Motivation*: Obviously, we could motivate participants who filled the questionnaire to play and finish the game. It is very important to understand, what the source of this motivation was. *Intrinsic* motivation basically means, a participant is motivated by the task itself while purely *extrinsic* motivation means, he is motivated by external influences. The questions in this part of the questionnaire are designed to determine the nature of participants' motivation. For the examined system intrinsic motivation would be the more sustainable one, as keeping extrinsic motivation up could be quite costly in the longterm (see Section 5.5.5).

5.5.4. Campaigning

The user study started on June 24th and ran until July 26th 2013. One of the hardest parts of any user study is getting enough participants. As it was already clear in the beginning

Figure 5.10.: GeoGame website

that we do not have the resources to do a large-scale experiment, with a participant group that is representative of the whole German population, we focused our marketing efforts around the university. A small 2-page flyer with some basic information about the game was created (see Appendix A.3). The flyer explains that reaching certain locations with a smartphone would earn the player a reward in the form of sweets. It does not mention that this is part of a user-study or that there will be a questionnaire to avoid biasing participants. Additionally, during the campaign we curated a GeoGame website (see Figure 5.10), that informed about the game and published news using a Twitter account.

The flyers have been distributed during the annual "TU Night" event where the university presents itself to interested citizens with a whole night of information and entertainment programs. This was also a chance to reach out to persons outside the university. During the user study we continued to deposit the flyers daily at one of the university's canteens. Early during the campaign the social-media team of Braunschweig noticed the experiment and posted some information on the city's official Facebook profile.

In the course of this user study the application has been installed on 219 different devices. Figure 5.11 shows, how the number of installations goes up at the beginning of the campaign, reaches a maximum in the middle of July and then starts to decrease. This shows that most participants have been convinced in the first part of the campaign, while near the end of the campaign people who finished the game uninstalled it.

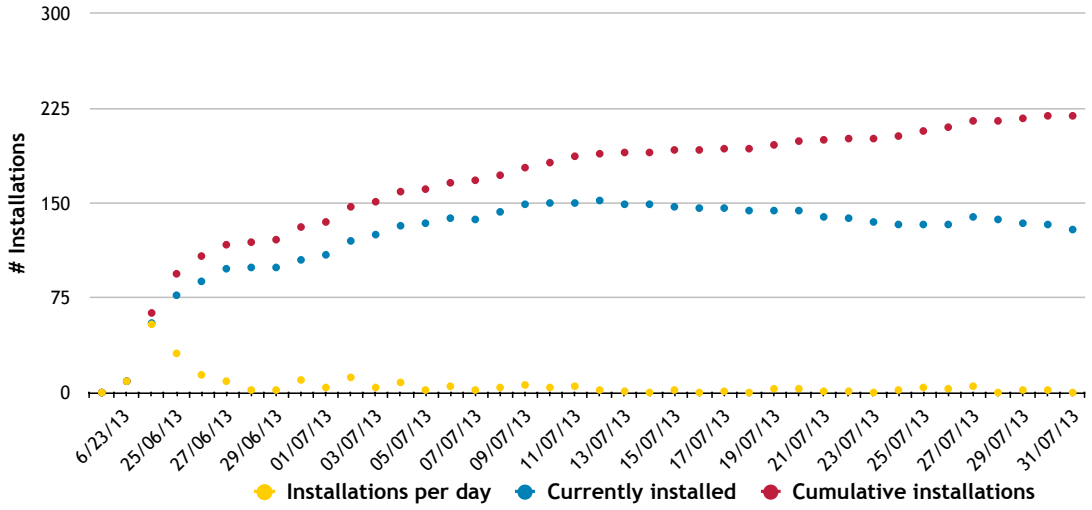


Figure 5.11.: GeoGame installs over time

5.5.5. Evaluation

The official playing phase for the GeoGame ran 2013 from June 24th until the 26th of July. After that date the application was pulled from the Google Play Store, but we gave players who already started the game a chance to finish it until the 1st of August. This extension was communicated by the application starting one week before the official end of the experiment.

In total, the GeoGame has been installed on 219 different devices. 72 devices (32.9%) completed at least one of the goals. This already shows, that there are a lot of people, who were willing to install the game, but probably immediately forgot about it. 48 (21.9 %) persons completed all 3 Braunschweig goals, which is a pretty good conversion rate from the 219 initial installs. This is also the relevant number for any deployed system that aims to leverage volunteers to transport data in a DTN. For this user study it was required that participants come to a university office at predefined times to get their reward and do the questionnaire. 31 persons did that (64.8% of the people who finished the game).

As we only gathered information from people installing and finishing the game, the results presented here do not provide any direct insight as to why some persons did not play or finish the game. However, as we will see, some ideas can be extrapolated from the answers of the finishing participants. Also, the amount of persons downloading and starting to play the game shows, that in principle it is possible to gather enough initial interest for deploying a feasible DTN system.

Demographics

Persons between 21 and 77 years completed the questionnaire. The average age was 29.5 years (median 26 years). 77.4% of the participants have been male and 22.6% female. As expected, most participants where associated with the university: We got 51.6% students and 32.3% research staff.

Smartphone Usage and Familiarity

We asked about the general proficiency of users with smartphones and location based applications. 58.1% of the participants said they "always" have their smartphones with them, and another 29% answered they carry their smartphone "most of the time" This is in line with other studies: A survey conducted 2013 in Germany concluded that 75% off all citizens never leave their homes without a mobile phone [109].

We also checked whether our participants had already an interest in location-based applications. The general question about the interest in location-based applications was rather inconclusive: 22.6% reported a "strong" interest, 45.2% a "moderate" interest, 25.8% "little" and 6.5% "no" interest. We also asked specifically about Geocaching, which is pretty popular in Germany. Only 6.5% of the participants said that they regularly hunt for caches. However, the remaining 93.5% have either tried Geocaching once or at least heard about it. This shows that our group was not particularly biased towards location-based games, but also that the concept is already well-known.

GeoGame Application Experience

We asked, how well the GeoGame app worked and what the user experience was. As the goal was to evaluate the feasibility of smartphone based DTN concept, we needed to make sure, the results are not biased by a bad implementation. Users were asked to use the German school grading system (1.0 for outstanding to 5.0 for insufficient) to rate the application. It scored a solid 1.6 for usability and 1.8 for functionality. The only problem mentioned to us have been occasional GPS location difficulties on some devices.

Difficulty

80.6% percent of the participants judged their familiarity with the city as "good" or "very good". 96.8% agreed that reaching the goals has been "easy" or "very easy". We asked which mode of transportation was used to reach the goals. Multiple answers could be chosen. A majority of people said they walked to the destinations (58.1%) or they used the bike (48.4%). Regarding the goals in Braunschweig 71% of the participants are near the university goal "daily" or "often", while for the goal in the Magni district this is only true for 25.9%. This is in line with the fact that most participants were university students or staff. Only one person said that he is "often" near the bonus goal in Hannover.

Bonus Goal

While we were not sure whether anyone would go 60 km to the bonus goal in Hannover, in fact 4 out of 31 (12.9%) players did go there. This is interesting, since only one participant (who did not complete the Hannover Bonus goal) claimed that he is often near that point. From the information available to participants before the game, it should have been quite clear, that despite the promised reward, going to Hannover for the sake of the game alone would probably not make any economic sense.

We asked the reasons for skipping or finishing the bonus point. Multiple answers have been possible. The main reasons for people not going to the bonus goal were "too much

distance" (71%) or "not enough time" (45%). Two persons said they did not feel like it, and the game was not entertaining enough. Of the four persons who finished the bonus goal all said they did it because they "wanted to". Two participants added they also did it, because they wanted to double their reward. Remember, that after completing the three goals in Braunschweig, the application offered participants to finish the bonus goal to double their reward. As we asked everybody to choose as many sweets as they deemed *appropriate* (see Section 5.5.5), it would have been up to the players to double their own reward. None of the four participants completing the Hannover goal lived in Hannover or had chosen the option "Because I am regularly in Hannover".

These results already give a first strong indication, that weighing the actual effort against the reward in a purely economic way is not the main motivation to reach the goals.

Trust

It is common knowledge that many applications and services invade the privacy of their customers. Fear of being spied on, or sharing valuable private data, can be prime concern for not adopting a new application or service. For this experiment it is important to note, that the user study falls directly in the time when the first Snowden documents regarding PRISM and the NSA came up [110]. While in June 2013, when this user study was conducted, this was still a non-topic in the US, in Germany it dominated media and public discussion very fast. Therefore, the results here might change if the study is repeated.

We asked, whether the participants believed that the GeoGame did not collect any personal data. 54.8% believed that claim, which means 45.2% were unsure or did not believe it. This shows a certain awareness for the problem, but maybe also some sort of resignation, since almost half of the people were not convinced that no data was collected, but still participated in the experiment.

We asked participants who were unsure regarding the claims that no personal data has been collected, what factors would inspire the most trust in an application or its creators. The first choice was "A renowned magazine reports about it" (29%) followed by "Many people use it" (9.7%).

Reward

After answering all previous sections there was a break in the questionnaire, where a participant gets his or her reward. Keep in mind that regarding the rewards there have been two related goals in this study: Will people accept and be happy with an (economically feasible) reward, such as sweets? And also what kind of reward and what amount would participants deem reasonable? This implies, we need to choose a reward *we* thought of being sufficient prior to the study. To get a clearer view what is acceptable by participants, and to avoid misjudging the appropriate amount, we adopted the following strategy: A full bowl of sweets and candy (see Figure 5.12) is presented and the participant is instructed to "take as much as is appropriate for finishing the game". The instructor would not say anything more. An interesting pattern emerged: A majority of people would first grab the whole bowl with a remark like "Ok then... as much as I want huh?". But in all those

Figure 5.12.: The reward bowl

cases, without any intervention on part of the instructor, the participant would put the bowl down and choose some amount. To our surprise, people really did not take much. Even though offered the opportunity to fully compensate their perceived costs, the average amount of sweets grabbed was $61.65g \pm 47g$. The largest amount was $201g$. We noticed, that many people tried to rationalize their choice like "Three goals, three pieces".

This indicates that rather small rewards might be enough to support a volunteer-driven DTN. However, we also asked how happy participants have been with their reward. While 41.9% considered themselves "very happy", 38.7% considered themselves only "somewhat happy", and the remaining 19.4% gave even lower scores. This shows, that despite the fact, that people where free to choose the amount of sweets, when asked directly some also felt that it was probably not appropriate or enough considering their effort. This may interpreted in such a way, that physical rewards may not be the right choice to support such as system. With a physical thing of concrete value it is always easily possible to rationally weigh the profit against the investment.

We asked which kind of reward participants would prefer. The options were "coupons", "money", "sweets" or "others" (with a free text field to specify). Only 14.3% have chosen "money". Noteworthy recurring mentions in the "other" category have been "leader boards", "achievements" and "more entertainment/story in the game". So even when given the ability to freely choose some substantial reward, many participants seem to feel virtual goods are enough or more appropriate *and* desirable. This also shows that people accepted even the rather barebone GeoGame as a game and, despite the fact that a reward was promised at the beginning, did not view it as some kind of service they need to be paid for. This idea

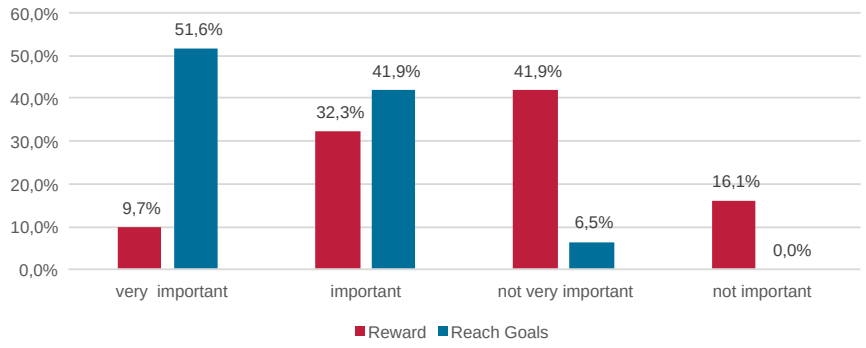


Figure 5.13.: Importance of rewards vs. achieving goals

can be substantiated when we compare the questions "How important was the reward?" and "How important was it to reach the goals?" as can be seen in Figure 5.13.

These results indicate, that a community supported smartphone-based DTN network can actually be deployed and operated economically. To get some more information whether this would be viable in the long term, we look at the results from determining the motivation of participants in the following section.

Intrinsic and Extrinsic Motivation

Relying on smartphone users to form a DTN is technically possible. Whether it is feasible and sustainable depends on the *motivation* of the participants. There are many criteria and classifications for motivation. Most important for this user study is the difference between *intrinsic* and *extrinsic* motivation [111].

If somebody is intrinsically motivated to do something, he does it for the fulfillment gathered from the action itself. For example, studying something, purely because it is interesting. Conversely, if somebody is extrinsically motivated to do something, he performs an action as a means to an end: For example, studying something solely to get a good grade. The outcome is separated from the action itself. The desired goal is to get a good grade. One will do whatever necessary to reach the goals with an appropriate, preferably minimal, effort. Often the employee in an employer-employee relationship is another prime example for extrinsic motivation.

This simple taxonomy of extrinsic motivation is extended by Self-Determination Theory (SDT) [112]. One can make a distinction, how internalized the causes for a motivation are. In this study we differentiate between *externally regulated behavior*, which is equivalent to the example of extrinsic motivated behavior above: A reward (or fear of punishment) is the reason for doing something, the controlling factor is perceived as external by the acting individual. Another case is *identified regulation* or the even stronger *integrated regulation*, where the cause of motivation is still external, but conducting an action or behaving in a certain way is considered to be of *personal importance*. The reasons for doing something are at least partially perceived as internal. An example where motivations might be internalized is structural violence as introduced by Galtung [113]. These are social, cultural or political

	Intrinsic (A)	Identified (B)	External (C)	Amotivation (D)
Average	17.65 ± 4.22	9.71 ± 3.59	7.94 ± 3.31	10.45 ± 5.10
Median	18.0	11.0	7.0	9.0
Percentage	73%	54%	44%	44%

	Sustainable (A+B)	**Unsustainable (C+D)**
Average	27.35 ± 7.07	18.38 ± 6.80
Median	29.0	18.0
Percentage	65%	43%

Table 5.2.: Participants motivation

conventions encouraging a certain behavior. Finally, there is also *Amotivation*, if somebody is not compelled, neither by external nor internal forces, to do something.

For the system examined in this user study intrinsic or at least internalized motivation is preferable, because it is clearly needed to make the system economically feasible. With externally regulated extrinsic motivation, people would compare the reward with their effort and desire a payment that they feel covers their costs. Just considering the traveling distance needed to reach the three goals within Braunschweig goals and the minimum time, which is around 15 min, would demand a reward equal to several EUR when basing the calculation on transportation fees and a minimum wage for the time needed. Some amount of intrinsic or internalized motivation is needed to be able to motivate enough data carriers at a feasible cost.

In this part of the questionnaire participants are presented with a set of 16 statements and need to choose their degree of agreement with each statement on a 6 point Likert scale. The statements can be classified into four categories: Agreement with four statements indicates intrinsic motivation. Three statements represent identified or integrated regulation, which is still a usable source of motivation for the system. Three statements describing external regulation and four statements describing amotivation are problematic: If the majority of participants would exhibit a stronger agreement with those statements, it means that the studied system would not be able to operate sustainably. Two statements have been assigned to the aforementioned classes after correlation of the answers to the a-priori classified statements. One of those falls into the intrinsic class while the other was correlated with external motivation. When coding the Lickert scale from 6 points for strong agreement to one point for disagreement the results of the answers to this question block are as shown in Table 5.2.

The percentage row in Table 5.2 makes it easer to compare the classes: To get 100% for a given class, a participant would need to chose the highest rating on the Likert scale for all statements belonging to that class. It can be seen, that the average participant is 73% intrinsically motivated, which is an encouraging result. External or amotivation

tendencies are both at 44%. When comparing the desirables causes for motivation (intrinsic to identified) with the unsustainable ones (external or amotivated), we can see that on average the desirable causes beat the unsustainable ones by 65% to 43%. This can be interpreted as proof, that a volunteer-driven smartphone-based DTN system could work. The results are especially good, considering the implemented system was rather barebone and did not contain any advanced or more exciting challenges like those mentioned in 5.3.3. The amotivation tendency is larger than we expected. From the answers, we assume there have been a number people who just played the game to help with our study without taking much real interest in it. The reason they mostly ended up in the amotivation instead of the external class, or not at all due to not finishing the game, is most likely the low perceived difficulty of the challenges posed by the game.

Out-Of-Band Answers

Apart from the questionnaire people have talked to us during the playing phase or after the questionnaire. Some people have also used the unrestricted commentary fields of the questionnaire to put some additional thoughts. In this Section we want to mention some things which came up.

It became clear immediately that despite the rather simple presentation people primarily enjoyed the game-play aspects of the study. Repeatedly, a central high score list or leader boards have been mentioned as improvement to the system. This is also illustrated by another "incident": Shortly after the game phase started, we made a post on Twitter informing that a player reached all goals within 16 minutes. At that point our intention was just to make the Twitter feed more interesting. Keep in mind, since we did not have any data about the players, the post could not even mention any name or nickname, but merely stated "a player". However, around one week later a participant turned up for his reward and questionnaire and asked about his time, because he was pretty sure his time must have been faster, but he was disappointed that did not see a Twitter post about it. It turned out his time was indeed faster (14 minutes). This highlighted an exploitable element of competition which we did not anticipate.

Another participant mentioned that he was able catch the university point (see Figure 5.7a) in Braunschweig while driving past it with his car. This point might be worth considering by people designing VANET applications: By choosing locations accordingly this kind of games could provide some sorely needed applications and additional benefit for early adopters of VANET technology. Currently the industry faces the chicken-and-egg problem that most VANET scenarios related to traffic and safety only work well once a high market penetration has been reached. However, putting out some stations for a GeoGame-like entertainment application in some metropolitan areas would be pretty cheap for car manufacturers while at the same time it would be immediately usable by any equipped car. This is in contrast to other envisioned VANET applications that have the chicken-and-egg problem of relying on a high density of suitably equipped cars.

Repeatedly, people mentioned that they would play again, but only if the task is varied. Mostly people had no interest to walk to the exact same locations again, but with another

set of locations or a varied task, such as time pressure, many people said they would play again.

Limitations

There are some limitations in this study. As it only includes a limited number of people, the results are not representative. However, due to the variance and conclusiveness of the answers, we are quite sure that this study disclosed some tendencies which would also hold up in a large-scale experiment. A problem is that the participant group was largely centered around university (see 5.5.5), and therefore the results only apply to this group. However, when implementing an incentive-based DTN system, it is not important that *everybody* can be motivated to support it, but rather that *enough* people support it. Therefore, we can conclude that the social stratum represented by university staff and students is a good target for such a system. Furthermore, the participants from outside the university context provided basically the same answers as the university-related participants. Hence, we have some confidence that the tendencies identified by this study are generally applicable.

There is one question which any user study like this can not completely answer: For many participants it might have been interesting to play because of the novelty of the game. What is really needed to keep that interest up, can not directly be derived from this user study, although as we got some pointers from the received answers: Participants expect themselves to have continued interest, should varying and interesting tasks or more competitive elements be offered.

5.6. Related Products

While we mentioned some related research within the scientific community in the introduction, we also want to mention some related commercial projects. A comparable user experience is provided by SCVNGR[10], a startup company offering location-based social games quite similar to the challenges of the GeoGame. A web front-end can be used to create challenges, sending people to locations where they can be asked to solve various questions. While virtually non-existent in Europe, the application is more widespread in North America. The company has acquired some venture capital and tries to sell the service to businesses for marketing or universities offering orientation rallies. SCVNGR is purely GPS-based, so that in its current form it cannot support any form of networking.

In 2008 Jordanian-based company Javna[11] introduced a system called MobiAd[12] offering localized “interactive” advertisements. Rewards for users are rebate coupons and MobiAd loyalty points. While the project focuses on cooperation with Jordanian-based communication provider Umniah[13] and the provider Zain[14] from Kuwait, since 2013 it can be downloaded and used by anyone from the Google Play store.

[10] `http://www.scvngr.com`
[11] `http://www.javna.com`
[12] `http://mobiadhome.com`, unrelated to she similarly named system presented [106])
[13] `http://www.umniah.com/`
[14] `http://www.zain.com`

One of the most successful location-based games is the relatively new Ingress[15] from Google. Ingress is an augmented reality game where members of two different factions need to physically visit so called "portals", which usually are landmarks, to gain control over them. After transitioning out of a semi-closed beta phase to an open beta in October of 2013, Ingress has quickly gathered players from all over the world. An unofficial community-driven study about Ingress players is available at [114], but unfortunately it did not ask directly what motivates players to play. In summer 2013 Google started experimenting with advertisement, by putting "portals" into the venues of advertisement partners. The Spotawin system presented in Section 5.4.3 focuses on the advertising aspect and is very light on gaming elements.

We do not know, whether those companies ever did some unpublished studies analyzing the motives of their users. Without exception, all of them still need to prove whether they can be economically self-sustainable. However, these examples confirm the findings of this study: Providing an entertaining game can provide enough incentive to keep users moving between arbitrary locations with their phones. While SVNGR as well as Ingress have started trying to make money based on advertisements, the potential of a DTN network provided by the players has not been tapped so far. Theoretically, both systems could be extended easily to support the style of networking proposed in this chapter. The applications would only need to be extended to optionally connect to a local base station instead of just checking GPS coordinates.

5.7. Summary

While the previous chapters dealt with the question how to scale DTNs up and integrating a large number of mobile nodes into the network, in this chapter we took a look at the problem of enticing users to join such a system. To reach widespread adoption, besides special applications, general users need to join the network. One challenge is that the benefits of supporting a DTN network with one's individual device are not immediately clear. It is more like paying a tax: You will lose something, but have not any immediate feedback what you get in return. Even if the gains are substantial, we know how eager most people are to pay taxes. The solution proposed in this chapter builds on two components: To be able to offer users some direct benefit, we argue that advertising might provide suitable incentives, while not being more expensive or even more cost-effective than classical forms of advertising. We have seen that related systems realizing parts of the Smart Ad concept are already being introduced. Secondly, we argue to not focus on the data transportation capabilities of the system, but rather sell some form of entertainment to the users. If users are provided with a game, the data transportation can be seen as a requirement for the game that will neither interest nor deter users from the game. The implemented Smart Ad prototype has the additional benefit that it keeps users in control of their device, and thus energy usage and location sharing: Instead of running in the background the system relies on the user to visit certain locations and consciously interact

[15] http://www.ingress.com

wit the game.

The Smart Ad approach has been tested and a user study confirmed that it is easy to use and technically solid. An important question that remains is the long-term sustainability of such systems. Will users get bored after a while or is it possible to sustain a network over a longer time. We tackled the answer from two sides: With the help from psychologists, we developed a questionnaire that tries to identify what kind of motivation drove users to partake in the Smart Ad study. The study showed that people are more intrinsically than extrinsically motivated, meaning their goal was to beat the game rather than "work" for the reward. This is a necessary prerequisite for a sustainable DTN because literally paying enough to keep extrinsic motivation is not feasible. Furthermore, existing location-based games prove anecdotal evidence that this mode of playing is of interest to a large community. Most prominently, Google's Ingress has a large, and ever growing user base while the demands of the game are very similar to the Smart Ad prototype: Go to certain locations with your device and use the application there. Which locations are worthwhile to go, is determined by the game world. When DTN networking becomes more preeminent, and a large player such as Google sees an application for it, it is very likely that one of the existing game systems will be piggy-backed with DTN-capabilities.

6 Conclusions

DTNs provide a robust paradigm to create networks in environments that cannot bet networked using classical network paradigms. By adding the idea of store-carry-and-forward transportation of data a DTN can deal with scheduled or unexpected disruptions between nodes and cope with large delays between a sender and receiver. Hence, DTN technologies are already commonly used for IPN applications. With the global change of communication habits and devices towards mobile network access, today we live in a world with seemingly ubiquitous network access. However, for practical purposes, connectivity is quite spotty. Cellular connectivity might break or degrade to throughput levels not suitable for certain kinds of applications when moving through areas with different coverage and congestions levels. An individual user will only have access to a small fraction of the theoretically available Wi-Fi infrastructure, as different Wi-Fi infrastructure providers require individual contracts. At the same time, with the ubiquitous availability of smartphones as well as the development of commercial VANET technologies, mobile use-case are becoming increasingly more popular. Under these conditions having a DTN network layer and DTN-aware applications is an improvement to transparently work around the challenges of varying network connectivity.

In this thesis we examined whether the DTN paradigm, and its most advanced implementation, the BP, are able to form the basis for an interconnected DTN network that encompasses the whole Internet. We identified several issues that prevent scaling up the current DTN ecosystem and proposed solutions.

6.1. Contributions

The DTN idea already gained traction in a number of niche applications, most notably IPN, that require the properties offered by a DTN. A large body of research investigated how DTN systems can be applied to terrestrial communication problems: Building networks in areas without infrastructure, creating ad-hoc disaster relief networks, improve performance of cellular networks by offloading data, among others. The work so far dealt with domain-specific networks of limited size. In this thesis we made the point, that using a DTN technology stack is also beneficial for Internet applications. The question is how the DTN concept, and its most widespread and mature realization, the BP, can scale to networks up to the size of the Internet a userbase comparable to the size of contemporary popular web services. We identified and tackled several areas that needed work:

1. In the BP DTN EIDs have no mandatory hierarchy. Therefore, it is not generally possible to base a routing decision on an EID. A node usually knows only its immediate neighbors through some beacon-based discovery mechanism. Furthermore,

as many scenarios are non-deterministic, most available routing mechanisms are flooding-based multi-copy schemes. This made it infeasible to apply DTN to Internet scenarios. We solved this problem in two ways: We argue, that no DTN routing should be used when using the BP in the Internet, as IP address-based routing is already provided. This reduces the problem of applying DTN in the Internet to the problem of discovery: How can an EID be mapped to an IP address? We propose using a DHT for this, which is a reliable and flexible distributed data structure. Moreover, it can work without any additional infrastructure. We proposed, implemented and evaluated a baseline version of this approach using the BT DHT. By leveraging the existing BT network the naming service can operate in an efficient and resilient way even when only a few DTN nodes are deployed.

2. Opportunistic networks requires discovery mechanisms. This is a huge disadvantage for devices with limited energy resources, such as smartphones. Therefore, for battery-powered DTN nodes it might be beneficial to use well known data drop-off locations, when an infrastructure network is available instead of beaconing for neighbors. Normally this would require investing in a number of DTN routers hosted in some data centers. We presented an approach that allows using the existing mail server infrastructure. The introduction of the MCL allows mailservers to be used as set of always available DTN routers at no additional cost.

3. Scaling up the size of DTN networks and applications implies increased data volume in a DTN. With more bundles to manage, determining which bundles need to be exchanged on a contact becomes a non-trivial problem. It is infeasible to compare inventory lists of bundles and in opportunistic scenarios, where the history of a communication partner is not known, the synchronization has to work without prior state. Bloom filters provide many desirable properties when reconciling two sets: They are much smaller than lists, are efficient to calculate and relatively robust with regard to input parameters. However, the remaining false positive can prevent complete reconciliation between two sets and thus lead to bundle loss or larger delays in a DTN. We extended Bloom filters with a trie of hierarchical hashes that can eliminate false positives from the Bloom Filter. Practically, the presented approach has the same complexity and robustness as a vanilla Bloom filter, while removing the false positives. Compared to other approaches for exact set reconciliation the presented Bloom Trie is robust with regard to the chosen parameters and does not need the size of the difference set as input parameter, which is often hard to estimate correctly.

4. DTNs depend on the fact that enough data carriers are available. While technically the integration of personal smartphones into a DTN is desirable, the question is, how can users be encouraged to support a DTN with their personal devices. An incentive needs to entice users to participate while at the same time being economically feasible for the DTN operator. We developed a system that motivates users

primarily by presenting itself as a game. Investments into DTN infrastructure and small-scale incentives in the form of coupons can be financed by using the system as advertisement channel. The games and advertisement are independent of the actual data carrying functionality. The system is economically self-sustainable no matter there is data to be transported or not. To verify whether such a system can entice users, we implemented a prototype that required users to visit a number of places in Braunschweig and Hannover. This experiment was evaluated using a questionnaire that also assessed the motivational factors to gain an insight, whether it can be expected that enough users would continuously support such a system. The results indicate that a user-supported DTN is in fact a feasible way to operate a DTN economically.

6.2. Outlook

In this thesis we highlighted and proposed solutions to several challenges when adapting DTN to networks as large as the Internet. If the number of DTN nodes grow and the BT DHT service presented in Chapter 3 is widely implemented, the DTN network might be large enough to support its own DHT. In this case a system that is more closely tied to the requirements of the BP, implementing some more advanced features from the Nasdi concept can be deployed. While the set reconciliation presented in Chapter 4 provides a robust basis for bundle storage synchronization, the best set of parameters can only be fine-tuned once there is a large enough DTN network with real traffic patterns. As it is to be expected that nodes with varying capabilities and storage sizes meet, a preliminary negotiation phase determining the optimal Bloom filter parameters for a specific contact can optimize performance. Efficient on-demand creation of the Bloom filters will be an interesting challenge. In Chapter 5 we argued that existing location-based games can easily be extended with DTN capabilities. Once some big players see the potential, the question is whether this will result several incompatible networks, or whether it is possible to agree on an interoperable protocol such as the BP that can be used to create a unified DTN network. Considering the situation of instant messengers today and the failure of XMPP [115, 116] to establish a federated messaging ecosystem, this seems a challenging proposition

While in all areas further research and refinements will surely be done and the available implementations of the BP continue to mature, the question is, where is DTN headed in the long term? It is clear that, for a lack of alternatives, DTN technologies will continue to thrive for IPN applications. However, one might question the premise of this thesis, that – despite the advantages– DTNs are the natural course of evolution for mobile Internet-based applications, or whether instead the current technology stack is “good enough”.

However, in the medium-term it is quite certain that the IPN domain and classical Internet will converge. Barring any kind of global disaster, it is only a matter of time before a substantial population will be living in the planetary neighborhood with the moon and the mars being the first candidates for larger settlements. Of course these settlements will have some kind of Internet, and will expect to be able to access services

hosted on earth. The problem of bandwidth has largely been solved today: While Viking probes launched 1975 could communicate science data with up to 1000 bit/s [117], the Mars Reconnaissance Orbiter launched 2005 can transmit data back to earth with up to 5 Mbit/s[1]. Currently research is underway to use optical technologies for communication. These have the potential to reach several Gbit/s over interplanetary distances [118]. The Lunar Atmosphere and Dust Environment Explorer (LADEE) mission launched in 2013 carried an experimental laser communication system (LLCD – Lunar Laser Communication Demonstration), which provides a bandwidth of up to 622 Mbit/s to earth [119, 120].

While the bandwidth can be improved with new technologies, the latency can not be changed. Therefore, in the medium-term the Internet will need to adopt DTN technologies and modify applications to be DTN-aware. Will this still be the BP? We do not know, but considering that the IP protocol was first specified in 1981 [121] and is still going strong more than 30 years later, it is save to say that it is not unlikely that future DTN systems will have evolved from the BP as used today.

[1] `http://mars.jpl.nasa.gov/mro/`

A Appendix

A.1. BT-DHT Configuration Options

```
#####################################
# DHTNameService settings           #
#####################################

#
# Enable the DHT, if it was compiled
# Default is no
#
dht_enabled = yes

#
# Set the udp port, the DHT should working on
# Default is 9999
# If Port is 0, a random Port will be chosen for each run
#
#dht_port = 9999

#
# Here you can choose a static DHT ID, which is very common
# Default is none -> a random ID per run will be generated
#dht_id = <randomstring>

#
# Enables DHT on IPv4 socket
# Default is yes
#
#dht_enable_ipv4 = yes

#
# Enables DHT on IPv6 socket
# Default is yes
#
#dht_enable_ipv6 = yes
```

```
#
# Bind the DHT to a specific IPv4 Address
# Default is the any device
#
#dht_bind_ipv4 = 127.0.0.1

#
# Bind the DHT to a specific IPv6 Address
# Default is the any device
#
#dht_bind_ipv6 = ::1

#
# Specify the file, where the DHT can save all good nodes
# for faster restart on next session
# Default is no file, but it should be set
#
#dht_nodes_file = <filepath>

#
# Enable DNS Bootstrapping for the DHT
#
#dht_bootstrapping = yes

#
# DNS Bootstrapping by giving domain names of wellknown nodes
#dht_bootstrapping_domains = [domain] [...]
#
# Example:
#dht_bootstrapping_domains = dtndht.ibr.cs.tu-bs.de
#
# Default is an empty string
#dht_bootstrapping_domains =

#
# IP Bootstrapping from wellknown IP (and port) addresses of nodes
#dht_bootstrapping_ips = [ip [port]]; [ip [port]]; ...
#
# Example:
#dht_bootstrapping_ips = 192.168.0.1; 192.168.0.2 8888;
```

```
#
# Default is an empty string
#dht_bootstrapping_ips =

#
# Blacklist support of the DHT can be switch on and off
#
# Default is yes
#dht_blacklist = yes

#
# Announcing myself on the DHT
#
# Default is yes
#dht_self_announce = yes

#
# Minimum necessary rating of a DHT information
#
# The lowest rating is 0: the node information has been sent
# by only one DHT node
# The maximum rating is 10 (for single lookups) and means:
# 10 or more different DHT nodes sent the information
#
# If the rating of an incoming information is lower, it will be ignored
#
# Default is 1
#dht_min_rating = 1

#
# Allow announcing neighbours
#
# Default is yes
#dht_allow_neighbour_announcement = yes

#
# Allow all neighbours announce them to be neighbour to me
# For privacy reasons, you could turn this off
#
# Default is yes
#dht_allow_neighbours_to_announce_me = yes
```

```
#
# Ignoring the neighbour information sent by a node, found on the DHT
#
# Default is no
#dht_ignore_neighbour_informations = no
```

A.2. MCL Internet Draft

This is the Internet Draft describing the technical details of the mail convergence layer protocol introduced in Section 3.6.

DTN Research Group
INTERNET-DRAFT
Intended Status: Experimental
Expires: October 25, 2013

B. Gernert, S. Schildt
IBR, TU Braunschweig

April 23, 2013

Delay Tolerant Networking Email Convergence Layer Protocol
draft-gernert-dtnrg-mailcl-01.txt

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress".

This Internet-Draft will expire on October 25, 2013.

Abstract

This document describes the protocol for the Email-based Convergence (MCL) Layer for Delay Tolerant Networking (DTN).

Copyright and License Notice

Table of Contents

1. Introduction

This document describes the Mail-based convergence layer protocol for Delay Tolerant Networking (MCL). MCL is based on SMTP and IMAP. Delay Tolerant Networking is an architecture providing communications in challenging networking environments, including those with intermittent connectivity, long and/or variable delays, and high bit error rates. A more complete characterization of these networks and their respective challenges can be found in the Delay-Tolerant Network Architecture [RFC4838].

An important goal of the DTN architecture is to accommodate a wide range of networking technologies and environments. A protocol used for DTN communications is the Bundle Protocol (BP) [RFC5050], an application-layer protocol that is used to construct a store-and-forward overlay network. The Bundle Protocol requires the services of a "convergence layer adapter" (CLA) to send and receive bundles using some underlying network protocol.

This document describes one such convergence layer adapter that uses SMTP and IMAP to transmit Bundles between BP nodes. With the MCL a BP node can have a mailbox, allowing for asynchronous DTN communication across the Internet when communication partners are not online at the same time. This allows leveraging legacy Internet services, without the need to deploy a native BP router in the Internet.

The locations of the MCL and the BP in the Internet model protocol stack are shown in Figure 1.

1.1. Conventions Used in this Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119. [RFC2119]

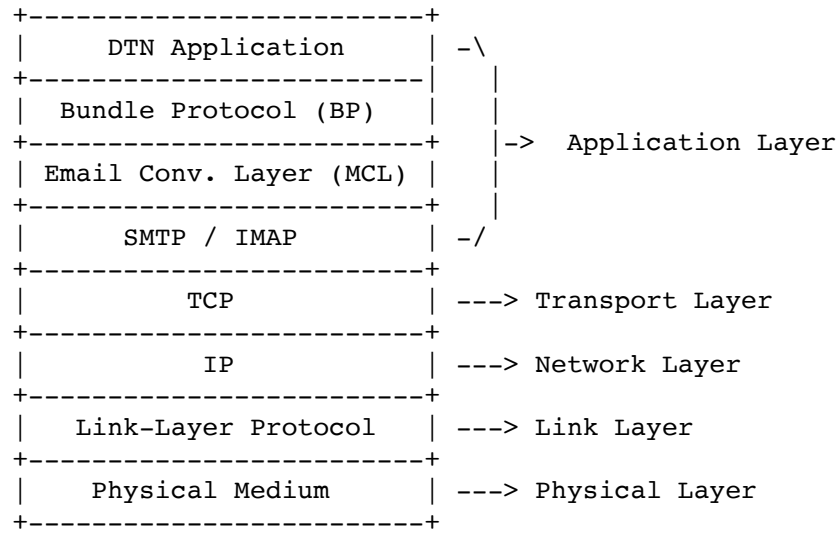


Figure 1: The location of the BP and the MCL
in the Internet protocol stack

2. Definitions

The following set of definitions are abbreviated versions of those which appear in the Bundle Protocol Specification [RFC5050]. To the extent in which terms appear in both documents, they are intended to have the same meaning.

Bundle
A bundle is a protocol data unit of the DTN bundle protocol.

Bundle payload
A bundle payload (or simply "payload") is the application data whose conveyance to the bundle's destination is the purpose for the transmission of a given bundle.

Bundle node
A bundle node (or simply a "node") is any entity that can send and/or receive bundles. The particular instantiation of this entity is deliberately unconstrained, allowing for implementations in software libraries, long-running processes, or even hardware. One component of the bundle node is the implementation of a convergence layer adapter.

Bundle endpoint and EID

A bundle endpoint (or simply "endpoint") is a set of zero or more bundle nodes that all identify themselves for BP purposes by the same URI [RFC3986], called a "bundle endpoint ID" (EID). The special case of an endpoint that never contains more than one node is termed a "singleton" endpoint; every bundle node must be a member of at least one singleton endpoint.

Extension blocks
Extension blocks are all blocks other than the primary and payload blocks.

3. Overview of the Protocol

This specification provides a means to exchange bundles through an e-mail server. It specifies how to encode bundles within a mail.

To be able to use the MCL a node needs its own email address and access to a mail server. A node will download received bundles from the mail server and use it to send bundles to other nodes supporting the MCL.

How to find the correct mail address for a BP EID is out of scope of this specification. The mail address for an EID may be configured statically, retrieved using existing discovery mechanism such as IPND or BT-DHT or, for MCL-only nodes, a new EID scheme encoding the mail address directly into the EID can be devised.

A bundle that has been successfully transmitted to a mail server will be considered delivered by the sending node. To ensure end-to-end acknowledgement of reception BP Status Reports can be used as normal.

3.1 Example Communication

Figure 2 shows the communication between two nodes using the MCL. Once node A's BP implementation comes into possession of a bundle destined for node B it will search for suitable forwarding opportunities. If node A gets to know B's MCL email address it will encode the bundle into a mail and submit it to its SMTP server.

Now the standard Internet mail system and protocols take over, finally delivering the bundle to the inbox of node B's MCL email address. Once B's MCL compliant BP implementation notices a new mail in its MCL email address' inbox, it will retrieve the mail and decode the contained bundle and deliver it to an application or forward it further according to the BP [RFC5050].

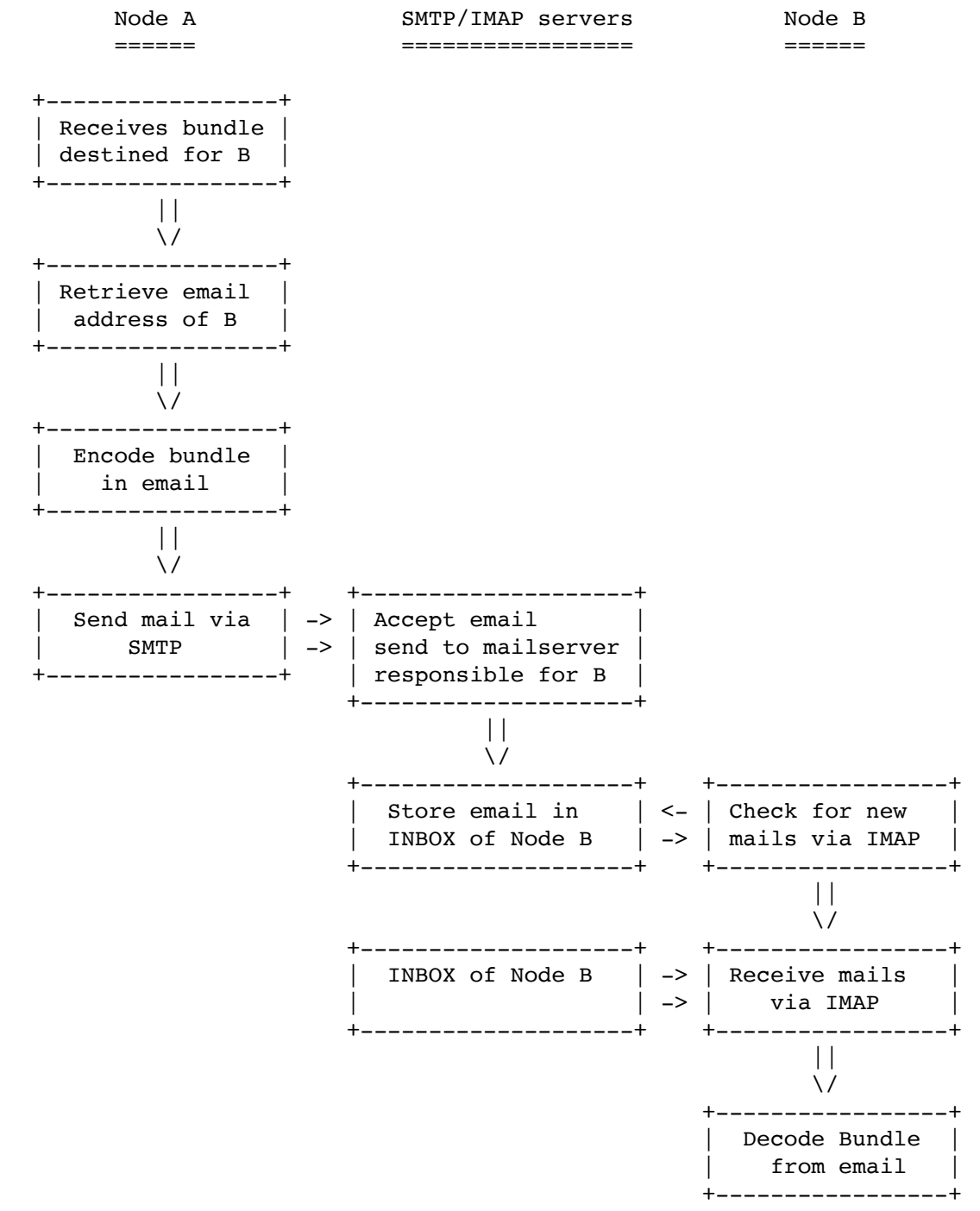


Figure 2: Communication between two nodes using MCL

4. Email Format

This section describes how to encode a bundle into a mail messages compliant with [RFC5322].

A mail consists of two parts: The header and the body. Additional data can be attached to a mail encoded as MIME parts ([RFC2045], [RFC2046], [RFC2047], [RFC2048], [RFC2049]).

A BP Bundle consists of the Primary Bundle Block, a Bundle Payload Block and an arbitrary number of Extension Blocks and administrative blocks (Bundle Status Reports, Custody Signals). The MCL encodes fields from the Primary Bundle Block in the mail headers. This allows a MCL implementation to decide whether to accept a bundle by only fetching the headers from the IMAP server without the need to download the whole bundle. The header fields from the Bundle Payload Block will also be written to the [RFC5322] mail header.

The mail header also contains references to the payload data as well as to all Extension Blocks and Administrative blocks. These blocks will be attached as MIME Parts. There must be one MIME part of the type application/octet-stream for each block.

The mails subjects field and the message body should contain some text explaining that this mail contains a bundle. Information about the bundle may be given in human-readable form.

Unless otherwise noted all fields have the same meaning content as described in [RFC5050]. Regardless of type, all fields are encoded as strings in the mail header. For INTs and SDNVs a decimal representation is used. Thus
The STRING "somestring" is encoded as "somestring"
The SDNV with the bit pattern 10000001 00001111 is encoded as "143"
The INT with the bit pattern 10000001 00001111 is encoded as "33039"

4.1 MCL mail headers

Table 1 shows the additional MCL headers that are not derived from [RFC5050].

Header-Field	Type	Optional
Bundle-EMailCL-Version	INT	
Bundle-Additional-Block	STRING	x

Table 1: MCL specific header fields

Bundle-EMailCL-Version:
Version of the MCL specification. A "1" denotes the version of the MCL described in this document. Other values are reserved for further use.

Bundle-Additional-Block:
If a bundle contains any blocks besides the Primary Bundle Block or the Payload Block, such as extension blocks, for each of these blocks a separate mail attachment must be created. Each attachment will have its own unique name. For each of these attachments one "Bundle-Additional-Block" header field must be created. Therefore this header field may occur multiple times in the header.

4.1 Encoding of the Primary Bundle Block

Table 2 shows the header fields of the Primary Bundle Block that will be put into the mail header. Except the fields marked as optional, all fields are required in a valid MCL mail.

Header-Field	Type	Optional
Bundle-Flags	INT	
Bundle-Destination	EID	
Bundle-Source	EID	
Bundle-Report-To	EID	
Bundle-Custodian	EID	
Bundle-Creation-Time	SDNV	
Bundle-Sequence-Number	SDNV	
Bundle-Lifetime	SDNV	
Bundle-Fragment-Offset	SDNV	x
Bundle-Total-Application-Data-Unit-Length	SDNV	x

Table 2: MCL header fields from the Primary Bundle Block

Bundle-Flags:
Processing flags for the Primary Bundle Block. This is a SDNV encoding the different flags as described in [RFC5050], which is limited to 19 bits. The MCL encodes an INT representation of this value.

Bundle-Destination:
The destination of the bundle. This is an EID in [RFC5050]. The MCL encodes the STRING representation of this value

Bundle-Source:
The source of the bundle. This is an EID in [RFC5050]. The MCL encodes the STRING representation of this value.

Bundle-Report-To:
The node to which status reports pertaining to the forwarding and delivery of this bundle are to be transmitted. This is an EID in [RFC5050]. The MCL encodes the STRING representation of this value.

Bundle-Custodian:
The node which is the current custodian of this bundle. This is an EID in [RFC5050]. The MCL encodes the STRING representation of this value.

Bundle-Creation-Time:
The creation time of the bundle. This is a SDNV in [RFC5050]. The MCL encodes an INT representation of this value.

Bundle-Sequence-Number:
The sequence number of the bundle. This is a SDNV in [RFC5050]. The MCL encodes an INT representation of this value.

Bundle-Lifetime:
The lifetime of the bundle. This is a SDNV in [RFC5050]. The MCL encodes an INT representation of this value.

Bundle-Fragment-Offset:
If this bundle is a fragment this header field must be set. This is a SDNV in [RFC5050], indicating the offset from the start of the original application data unit. The MCL encodes an INT representation of this value.

Bundle-Total-Application-Data-Unit-Length:
If this bundle is a fragment this header field must be set. This is a SDNV in [RFC5050], indicating the total length of the original application data unit of which this bundle's payload is a part. The MCL encodes an INT representation of this value.

4.2 Encoding the Bundle Payload Block

Encoding the Bundle Payload Block is similar to encoding the Primary Bundle Block. Table 3 list the header fields related to the Bundle Payload Block. The actual payload data will be attached as MIME part.

4.2.1 The Payload Block

Header-Fields	Value	Optional
Bundle-Payload-Flags	INT	
Bundle-Payload-Block-Length	SDNV	x
Bundle-Payload-Data-Name	STRING	

Table 3: Header fields from the Payload Block

Bundle-Flags:
Processing flags for the Payload Block. This is a SDNV in [RFC5050], which is limited to 19 bits. The MCL encodes an INT representation of this value.

Bundle-Payload-Block-Length:
Length of the payload data in bytes. This field is optional as it is possible to calculate the size directly from the attachment.

Bundle-Payload-Data-Name:
Name of the MIME part which contains the payload data.

4.3 Encoding Extension Blocks

Other Bundle Blocks (Extension Blocks) must be attached as MIME parts. The headers for Extension Blocks are listed in table 4. These headers must be encoded in the headers of the MIME part and not in the general mail header.

Every extension block MUST be referenced in the mail header using the Bundle-Additional-Block field.

Header-Fields	Value	Optional
Block-Type	STRING	
Block-Processing-Flags	INT	
Block-EID-Reference	EID	x

Table 4: MIME-part header fields for Extension Blocks

Block-Type:
Bundle Block type code as specified in [RFC5050].

Block-Processing-Flags:
Processing flags for the Payload Block. This is a SDNV encoding the different flags as described in [RFC5050], which is limited to 19 bits. The MCL encodes an INT representation of this value.

Block-EID-Reference:
Each extension block may include one or more reference EIDs. For each of these EIDs a header field "Block-EID-Reference" will be created. The value of this field is the string representation of the specific EID. As an extension block may contain more than one EID reference, this header field will occur multiple times.

4.4 Encoding a Status Report/Custody Signal Block

The Status Report/Custody Signal Block are contained within the payload of the bundle. A specific bundle processing control flag will be set by the BP implementation to indicate that the payload is containing such an administrative record. Therefore no additional measures need to be taken to transmit an administrative record with MCL.

5. Example

The following example shows a correctly encoded mail with just a payload block. This bundle comes from "dtn://second/eid" with the mail address "sender@server" and was delivered to "dtn://some/eid" with the mail address "recv@server". The payload can be found in the

```
attachment named "payload.data".

Return-path: <sender@server>
Envelope-to: recv@server
Delivery-date: Wed, 23 Jan 2013 19:44:25 +0100
From: sender@server
To: recv@server
Subject: Bundle for mail://sender@server
Bundle-EMailCL-Version: 1
Bundle-Flags: 144
Bundle-Destination: dtn://some/eid
Bundle-Source: dtn://second/eid
Bundle-Report-To: dtn:none
Bundle-Custodian: dtn:none
Bundle-Creation-Time: 412281870
Bundle-Sequence-Number: 1
Bundle-Lifetime: 3600
Bundle-Payload-Flags: 8
Bundle-Payload-Block-Length: 35
Bundle-Payload-Data-Name: payload.data
Content-Type: multipart/mixed;
 boundary="=_f-20r0xUuORzjAo2CVz1bGFWJK1irHf4t+jNIoYURaTVkAY6"

This is a multi-part message in MIME format. Your mail reader does
not understand MIME message format.
--=_f-20r0xUuORzjAo2CVz1bGFWJK1irHf4t+jNIoYURaTVkAY6


--=_f-20r0xUuORzjAo2CVz1bGFWJK1irHf4t+jNIoYURaTVkAY6
Content-Type: application/octet-stream
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=payload.data

VGVzdA==
--=_f-20r0xUuORzjAo2CVz1bGFWJK1irHf4t+jNIoYURaTVkAY6--
```

7. IANA Considerations

This document has no actions for IANA at the moment.

8. Security Considerations

To secure the transmission to and from an email server you can use SMTP and IMAP over TLS [RFC3207] and [RFC2595]. However it is still possible that an attacker sends manipulated mails to an inbox of one node. As the MCL can encode the full BP feature set, the Bundle Security protocol extensions [RFC6257] can be used to secure the system against malicious bundles.

9. References

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.

[RFC2047] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.

[RFC2048] Freed, N., Klensin, J., and J. Postel, "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", RFC 2048, November 1996.

[RFC2049] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples", RFC 2049, November 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2595] Newman, C., "Using TLS with IMAP, POP3 and ACAP", RFC 2595, June 1999.

[RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, February 2002.

[RFC4838] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and H. Weiss, "Delay-Tolerant Networking Architecture", RFC 4838, April 2007.

[RFC5050] Scott, K. and S. Burleigh, "Bundle Protocol Specification", RFC 5050, November 2007.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.

[RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, July 2011.

Authors' Addresses

Bjoern Gernert
Technische Uninversitaet Braunschweig
Institute of Operating Systems and Computer Networks
Muhlenpfordtstr. 23
38106 Braunschweig
Germany

Email: mail@bjoern-gernert.de

Sebastian Schildt
Technische Uninversitaet Braunschweig
Institute of Operating Systems and Computer Networks
Muhlenpfordtstr. 23
38106 Braunschweig
Germany

Phone +49 531 391 3285
Email: schildt@ibr.cs.tu-bs.de

A.3. Geo Game Flyer

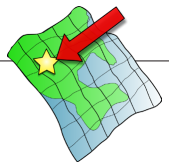
Technische Universität Braunschweig

Institut für Betriebssysteme und Rechnerverbund

GeoGame Braunschweig
Spiel mit uns!

Entdecke die Stadt!

Mit dem GeoGame Braunschweig lernst du nicht nur die Stadt besser kennen, sondern du hilfst auch noch der Wissenschaft - und dafür belohnen wir dich mit Süßigkeiten!



GeoGame ist ein Projekt des IBR der TU Braunschweig

Das GeoGame erklärt

IN DIE STADT

App runterladen reicht nicht, du musst in die Stadt! Drei Zielorte sind in Braunschweig zu erreichen - die App merkt sich, welche Ziele du erreichst und schreibt dir dafür Punkte gut. Wer beginnt, gewinnt!

GEOBASIERT

Du befindest dich schon auf dem Spielfeld! GeoGame spielt man mittels GPS draußen! Die App zeigt dir deine Position auf der Karte an und führt dich zu allen Zielen und deiner Belohnung!

BELOHNUNG

Die gibt's, wenn du alle drei Ziele erreicht hast. Das ist mit dem Fahrrad in 25 Minuten gemacht! Als Belohnung für deine Mühe gibt es köstliche Süßigkeiten!

Die Wissenschaft zählt auf dich!
Lass uns nicht hängen ...

Institut für Betriebssysteme und Rechnerverbund

Kontakt: Tim Lüdtke
t.luedtke@tu-braunschweig.de

SMARTAD.IBR.CS.TU-BS.DE

A.4. User Study Questionnaire

This are the questions from the Smart Ad user study questionnaire. This is only an export of the questions, the actual questionnaire was performed on web-based system, thus depending on the answers participants did not see all the questions. This export also shows which conditions triggered which questions to be shown. During the actual questionnaire these conditions have been checked automatically.

TU Braunschweig IBR GeoGame Spielerumfrage

Vielen Dank, dass du am IBR Spiel GeoGame teilgenommen hast!

Mit der folgenden Umfrage wollen wir deinen Eindruck vom Spielen festhalten.

Diese Umfrage benötigt weniger als 15 Minuten deiner Zeit.

Beantworte die Fragen möglichst schnell und intuitiv. Es gibt keine richtigen, oder falschen Antworten!

Diese Umfrage enthält 39 Fragen.

Allgemeines

Zu Beginn ein paar allgemeine Fragen. Du solltest das IBR GeoGame zu diesem Zeitpunkt abgeschlossen haben.

[]Wie war deine Spieler ID ? *

Bitte gib hier Deine Antwort ein:

Drücke in der App auf das (?) um deine Spieler ID anzuzeigen.

[]Wie alt bist du? *

Bitte gib hier Deine Antwort(en) ein:

Alter:

[]Was übst du für einen Beruf aus? *

Bitte wähle nur eine der folgenden Antworten aus:

- ○ Schüler(-in)
- ○ Student(-in)
- ○ wissenschaftliche(-r) Mitarbeiter(-in)
- ○ Sonstiges

[]Dein Geschlecht? *

Bitte wähle nur eine der folgenden Antworten aus:

- ○ Weiblich
- ○ Männlich

[]Wie gut kennst du dich in Braunschweig aus? *

Bitte wähle nur eine der folgenden Antworten aus:

- ○ sehr gut
- ○ relativ gut
- ○ mittelmäßig
- ○ kaum
- ○ gar nicht

[]Hast du dein Smartphone / Tablet ständig bei dir? *

Bitte wähle nur eine der folgenden Antworten aus:

- ○ immer
- ○ meistens
- ○ selten
- ○ sehr selten
- ○ Ich habe kein eigenes Smartphone

Geoapp

[]Kennst du "Geocaching"? *

Bitte wähle nur eine der folgenden Antworten aus:

- ○ Ja, ich mache es regelmäßig
- ○ Ja, ich habe mal mitgemacht
- ○ Ja, ich habe mal davon gehört
- ○ Noch nie davon gehört

[]Wie stark ist allgemein dein Interesse an Apps, die deine Position in die Handlung mit einbeziehen? *

Bitte wähle nur eine der folgenden Antworten aus:

- ○ stark
- ○ mittel
- ○ gering
- ○ sehr gering

Etwa bei einer Stadtführer-App, Foursquare oder im Rahmen eines Spiels wie bei Google Ingress.

[]Welche Eigenschaften sind dir allgemein bei Apps wichtig? *

Bitte nummeriere jede Box in der Reihenfolge Deiner Präferenz, beginnend von 1 bis 5

- ☐ Design
- ☐ Fehlerfreiheit
- ☐ Spaß
- ☐ Funktionalität
- ☐ Datenschutz

Anreiz

[]Über welchen Weg hast du vom IBR GeoGame erfahren? *

Bitte wähle **alle** Punkte aus, die zutreffen:

- ☐ Jemand hat mir davon erzählt
- ☐ In einer E-Mail wurde zum Spiel eingeladen
- ☐ Ich habe einen QR-Code abgescannt
- ☐ Ich habe über das Internet davon erfahren
- ☐ Sonstiges: _____

[]Was war deine Motivation um beim IBR GeoGame mitzumachen und wie wichtig war diese? *

Bitte wähle die zutreffende Antwort aus:

	sehr wichtig	wichtig	wenig wichtig	unwichtig
Belohnung	○	○	○	○
Neugier	○	○	○	○
Ich wurde von jemandem gebeten	○	○	○	○
Weil mein/e Freund/in auch mitgemacht hat	○	○	○	○

Bewerte bitte jede der vier Zeilen

Ziele

[]Welche Verkehrsmöglichkeiten hast du zum Erreichen der Ziele in Braunschweig benutzt? *

Bitte wähle **alle** Punkte aus, die zutreffen:

- ☐ zu Fuß
- ☐ Fahrrad
- ☐ Bus
- ☐ Strassenbahn
- ☐ Auto / Motorrad
- ☐ Sonstiges:

[]Kommst du an einem oder mehreren Zielorten des IBR GeoGame oft vorbei? *

Bitte wähle die zutreffende Antwort aus:

	Täglich	Oft	Selten	Nie
Uninähe (Rebenring)	○	○	○	○
Magniviertel (Magnitor)	○	○	○	○
am Stadtkern (Friedrich-Willhelm-Straße)	○	○	○	○
Bonuspunkt (Hannover)	○	○	○	○

[]Wie schwer fandest du beim IBR GeoGame Folgendes: *

Bitte wähle die zutreffende Antwort aus:

	sehr gut	gut	mittel	schlecht
Erreichbarkeit der Ziele	○	○	○	○
Bedienung der App	○	○	○	○
Funktionalität der App	○	○	○	○

Mit Erreichbarkeit ist gemeint, wie schwer es war die Spielziele zu erreichen.

Mit Bedienung der App ist gemeint, wie gut sich die App von dir bedienen lies.

Mit Funktionalität der App ist gemeint, wie gut die App technisch funktioniert hat.

[]

Welche Schwierigkeiten hattest du bei der Erreichbarkeit der Ziele?

Beantworte diese Frage nur, wenn folgende Bedingungen erfüllt sind:
Antwort war 'schlecht' *oder* 'mittel' bei Frage '14 [ZIELE_Schwierigkeit]' (Wie schwer fandest du beim IBR GeoGame Folgendes: (Erreichbarkeit der Ziele)) *und* Antwort war 'schlecht' *oder* 'mittel' bei Frage '14 [ZIELE_Schwierigkeit]' (Wie schwer fandest du beim IBR GeoGame Folgendes: (Erreichbarkeit der Ziele)) *und* Antwort war 'schlecht' *oder* 'mittel' bei Frage '14 [ZIELE_Schwierigkeit]' (Wie schwer fandest du beim IBR GeoGame Folgendes: (Erreichbarkeit der Ziele))

Bitte gib hier Deine Antwort ein:

Eine kurze Beschreibung hilft uns, dieses Problem zu beheben.

[]

Welche Schwierigkeiten hattest du mit der Bedienung der App?

Beantworte diese Frage nur, wenn folgende Bedingungen erfüllt sind:
Antwort war 'mittel' *oder* 'schlecht' bei Frage '14 [ZIELE_Schwierigkeit]' (Wie schwer fandest du beim IBR GeoGame Folgendes: (Bedienung der App)) *und* Antwort war 'mittel' *oder* 'schlecht' bei Frage '14 [ZIELE_Schwierigkeit]' (Wie schwer fandest du beim IBR GeoGame Folgendes: (Bedienung der App))

Bitte gib hier Deine Antwort ein:

Eine kurze Beschreibung hilft uns, dieses Problem zu beheben.

[]Welche Schwierigkeiten hattest du mit der Funktionalität der App?

Beantworte diese Frage nur, wenn folgende Bedingungen erfüllt sind:
Antwort war 'schlecht' *oder* 'mittel' bei Frage '14 [ZIELE_Schwierigkeit]' (Wie schwer fandest du beim IBR GeoGame Folgendes: (Funktionalität der App)) *und* Antwort war 'schlecht' *oder* 'mittel' bei Frage '14 [ZIELE_Schwierigkeit]' (Wie schwer fandest du beim IBR GeoGame Folgendes: (Funktionalität der App))

Bitte gib hier Deine Antwort ein:

Eine kurze Beschreibung hilft uns, dieses Problem zu beheben.

(Etwa: App ließ sich nicht starten, blieb immer hängen, stürtze bei XY ab, fand keine Position)

Bonuspunkt

Diese Seite bezieht sich auf das Bonusziel, das du optional in Hannover absolvieren konntest.

[]Hast du den Bonuspunkt absolviert? *

Bitte wähle nur eine der folgenden Antworten aus:

- ○ Ja
- ○ Nein

[]Aus welchen Gründen hast du den Bonuspunkt nicht absolviert? *

Beantworte diese Frage nur, wenn folgende Bedingungen erfüllt sind:
Antwort war 'Nein' bei Frage '18 [BonusJaNein]' (Hast du den Bonuspunkt absolviert?)

Bitte wähle **alle** Punkte aus, die zutreffen:

- ☐ Entfernung (Das Bonusziel war zu weit weg)
- ☐ Unkenntnis (Ich wusste nicht, dass das Bonusziel die Belohnung verdoppelt)
- ☐ Zeitmangel (Ich hätte das Bonusziel gerne gemacht, hatte aber nicht genügend Zeit)
- ☐ keine Lust mehr (Das Spiel hat mir keinen Spaß gemacht)
- ☐ Sonstiges: [ ]

[]Aus welchen Gründen hast du den Bonuspunkt absolviert? *

Beantworte diese Frage nur, wenn folgende Bedingungen erfüllt sind:
Antwort war 'Ja' bei Frage '18 [BonusJaNein]' (Hast du den Bonuspunkt absolviert?)

Bitte wähle **alle** Punkte aus, die zutreffen:

- ☐ Ich fahre sowieso oft nach Hannover
- ☐ Ich wohne in Hannover
- ☐ Dank meines Studententickets konnte ich kostenlos nach Hannover fahren
- ☐ Ich wollte meine Belohnung verdoppeln
- ☐ Ich hatte Lust darauf
- ☐ Sonstiges: [ ]

[]Welche Verkehrsmöglichkeiten hast du zum Erreichen des Bonusziels benutzt? *

Beantworte diese Frage nur, wenn folgende Bedingungen erfüllt sind:
Antwort war 'Ja' bei Frage '18 [BonusJaNein]' (Hast du den Bonuspunkt absolviert?)

Bitte wähle **alle** Punkte aus, die zutreffen:

- ☐ zu Fuß
- ☐ Fahrrad
- ☐ Bus
- ☐ Strassenbahn
- ☐ Zug
- ☐ Auto / Motorrad
- ☐ Sonstiges:

Sicherheit

[]"Im Rahmen dieses Spiels wurden keine Bewegungsprofile oder personenbezogenen Daten von dir aufgezeichnet." Glaubst du diese Aussage? *

Bitte wähle nur eine der folgenden Antworten aus:

- ○ Ja
- ○ Unsicher
- ○ Nein

[]Aus welchem Grund glaubst du die Aussage? *

Beantworte diese Frage nur, wenn folgende Bedingungen erfüllt sind:
Antwort war 'Ja' bei Frage '22 [AkzepSicherheit]' ("Im Rahmen dieses Spiels wurden keine Bewegungsprofile oder personenbezogenen Daten von dir aufgezeichnet." Glaubst du diese Aussage?)

Bitte wähle **alle** Punkte aus, die zutreffen:

- ☐ Datensicherheit interessiert mich nicht
- ☐ Ich bin von der Seriösität dieser App überzeugt
- ☐ Ich bin von der Seriösität der Entwickler überzeugt
- ☐ Solche Daten sind für diese Anwendung gar nicht relevant
- ☐ Die Aufzeichnung wäre technisch gar nicht möglich
- ☐ Sonstiges: _____

[]Aus welchem Grund glaubst du die Aussage nicht? *

Beantworte diese Frage nur, wenn folgende Bedingungen erfüllt sind:
Antwort war 'Unsicher' *oder* 'Nein' bei Frage '22 [AkzepSicherheit]' ("Im Rahmen dieses Spiels wurden keine Bewegungsprofile oder personenbezogenen Daten von dir aufgezeichnet." Glaubst du diese Aussage?) *und*
Antwort war 'Unsicher' *oder* 'Nein' bei Frage '22 [AkzepSicherheit]' ("Im Rahmen dieses Spiels wurden keine Bewegungsprofile oder personenbezogenen Daten von dir aufgezeichnet." Glaubst du diese Aussage?)

Bitte wähle **alle** Punkte aus, die zutreffen:

- ☐ Es ist nur so ein Gefühl
- ☐ Die App ist nicht seriös
- ☐ Die Entwickler sind nicht seriös
- ☐ Ich habe keine Möglichkeit dies zu überprüfen
- ☐ Sonstiges: _____

[]Wie sehr würden dich die folgenden Punkte von der Datensicherheit der App überzeugen? *

Beantworte diese Frage nur, wenn folgende Bedingungen erfüllt sind:
Antwort war 'Unsicher' *oder* 'Nein' bei Frage '22 [AkzepSicherheit]' ("Im Rahmen dieses Spiels wurden keine Bewegungsprofile oder personenbezogenen Daten von dir aufgezeichnet." Glaubst du diese Aussage?) *und* Antwort war 'Unsicher' *oder* 'Nein' bei Frage '22 [AkzepSicherheit]' ("Im Rahmen dieses Spiels wurden keine Bewegungsprofile oder personenbezogenen Daten von dir aufgezeichnet." Glaubst du diese Aussage?)

Bitte nummeriere jede Box in der Reihenfolge Deiner Präferenz, beginnend von 1 bis 4

- [ ] Eine Fachzeitschrift müsste positiv darüber berichten
- [ ] Die App müsste von vielen benutzt werden
- [ ] Ein/e Freund/in müsste sie mir empfehlen
- [ ] Die App müsste von einem renommierten Unternehmen/Entwickler kommen

Ordne die Punkte nach deiner persönlichen Wichtigkeit an. (Das Wichtigste nach oben)

[]Im App-Beschreibungstext im PlayStore wird erklärt, wozu die GeoGame-App welche Berechtigungen benötigt. Ist dies für dich von Interesse? *

Bitte wähle nur eine der folgenden Antworten aus:

- ○ Ja
- ○ Nein

VorBelohnung

Du bekommst jetzt endlich deine Belohnung!

[]*Du bekommst jetzt endlich deine Belohnung! Sag bescheid, dass du bis hierhin gekommen bist.*

Belohnung

Nun folgen einige Fragen zur Belohnung:

[]Wieviele Süßigkeiten hast Du dir ausgesucht? *

Bitte gib hier Deine Antwort ein:

Gramm

[]Wie wichtig ist dir die Belohnung beim IBR GeoGame? *

Bitte wähle nur eine der folgenden Antworten aus:

- ○ sehr wichtig
- ○ wichtig
- ○ wenig wichtig
- ○ gar nicht wichtig

[]Wie wichtig war es dir, alle Ziele zu erreichen? *

Bitte wähle nur eine der folgenden Antworten aus:

- ○ sehr wichtig
- ○ wichtig
- ○ wenig wichtig
- ○ gar nicht wichtig

[]Würdest du das IBR GeoGame nochmals spielen? *

Bitte wähle nur eine der folgenden Antworten aus:

- ○ ja
- ○ eventuell
- ○ nein

[]Welche Anreize müssten für dich gegeben sein, damit du das IBR GeoGame nochmal spielen würdest? *

Beantworte diese Frage nur, wenn folgende Bedingungen erfüllt sind:
Antwort war 'eventuell' *oder* 'nein' bei Frage '31 [nochmal]' (Würdest du das IBR GeoGame nochmals spielen?) *und* Antwort war 'eventuell' *oder* 'nein' bei Frage '31 [nochmal]' (Würdest du das IBR GeoGame nochmals spielen?)

Bitte wähle **alle** Punkte aus, die zutreffen:

- ☐ Es müsste mehr Zielpunkte geben
- ☐ Ich müsste das Spiel mit meinen Freunden spielen können
- ☐ Es müsste eine höhere Belohnung geben
- ☐ Sonstiges:

Belohnung2

[]Wie sehr freust du dich über Folgendes: *

Bitte wähle die zutreffende Antwort aus:

	sehr	etwas	wenig	gar nicht
Die erfolgreiche Erledigung der Aufgabe	○	○	○	○
Die Höhe der Belohnung	○	○	○	○

[]Empfindest Du deine Belohnung als angemessen, im Verhältnis zu deinem Aufwand? *

Beantworte diese Frage nur, wenn folgende Bedingungen erfüllt sind:
Antwort war 'sehr wichtig' *oder* 'wichtig' *oder* 'wenig wichtig' bei Frage '29 [wichtig Belohnung]' (Wie wichtig ist dir die Belohnung beim IBR GeoGame?) *und* Antwort war 'sehr wichtig' *oder* 'wichtig' *oder* 'wenig wichtig' bei Frage '29 [wichtig Belohnung]' (Wie wichtig ist dir die Belohnung beim IBR GeoGame?) *und* Antwort war 'sehr wichtig' *oder* 'wichtig' *oder* 'wenig wichtig' bei Frage '29 [wichtig Belohnung]' (Wie wichtig ist dir die Belohnung beim IBR GeoGame?)

Bitte wähle nur eine der folgenden Antworten aus:

- ○ Mehr als angemessen
- ○ Angemessen
- ○ Nicht angemessen

[]Welche Art von Belohnung würdest Du dir zukünftig wünschen? *

Bitte wähle **alle** Punkte aus, die zutreffen:

- ☐ Gutscheine / Rabatt-Coupons
- ☐ Geld
- ☐ Süßigkeiten
- ☐ Sonstiges: _____

[]Würdest du auch am IBR GeoGame teilnehmen, wenn es anstelle einer sicheren Belohnung die Möglichkeit einer höheren Belohnung durch eine Verlosung geben würde? *

Bitte wähle nur eine der folgenden Antworten aus:

- ○ Ja
- ○ Unsicher
- ○ Nein

[]Was wäre für dich persönlich eine erstrebenswerte Belohnung?

Beantworte diese Frage nur, wenn folgende Bedingungen erfüllt sind:
Antwort war 'Unsicher' *oder* 'Ja' bei Frage '36 [Verlosung]' (Würdest du auch am IBR GeoGame teilnehmen, wenn es anstelle einer sicheren Belohnung die Möglichkeit einer höheren Belohnung durch eine Verlosung geben würde?) *und* Antwort war 'Unsicher' *oder* 'Ja' bei Frage '36 [Verlosung]' (Würdest du auch am IBR GeoGame teilnehmen, wenn es anstelle einer sicheren Belohnung die Möglichkeit einer höheren Belohnung durch eine Verlosung geben würde?)

Bitte gib hier Deine Antwort ein:

[]Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken.

Ich habe am Spiel teilgenommen, ... *

Bitte wähle die zutreffende Antwort aus:

	stimmt völlig					stimmt nicht
... weil ich denke, dass das Spiel interessant ist	○	○	○	○	○	○
... weil ich es für mich selbst mache	○	○	○	○	○	○
... weil es von mir erwartet wird	○	○	○	○	○	○
Es gibt sicherlich gute Gründe für das Spiel, aber ich sehe diese nicht	○	○	○	○	○	○
... weil ich denke, das Spiel ist angenehm	○	○	○	○	○	○
... weil ich denke, das Spiel ist gut für mich	○	○	○	○	○	○
... weil es etwas ist, was ich zu tun habe	○	○	○	○	○	○
Ich spiele das Spiel, bin mir aber nicht sicher, ob es das wert ist	○	○	○	○	○	○
... weil es einfach Spaß macht	○	○	○	○	○	○
... weil es meine persönliche Entscheidung war	○	○	○	○	○	○
... weil ich keine Wahl hatte	○	○	○	○	○	○
Ich weiß nicht, was das Spiel mir bringt	○	○	○	○	○	○
... weil ich mich gut fühle, wenn ich das Spiel spiele	○	○	○	○	○	○
... weil ich glaube das Spiel ist wichtig für mich	○	○	○	○	○	○
... weil ich fühle, dass ich es zu tun habe	○	○	○	○	○	○
Ich spiele das Spiel, bin mir aber nicht sicher ob ich es sinnvoll ist es nochmal zu spielen	○	○	○	○	○	○

[]Möchtest du den Entwicklern des IBR Spiels noch etwas mitteilen?

Bitte gib hier Deine Antwort ein:

Vielen Dank, dass du am IBR Spiel GeoGame teilgenommen hast! Lass dir deine Süßigkeiten schmecken!

Absenden der Umfrage.
Vielen Dank für die Beantwortung des Fragebogens.

A.5. User Study Data

This is the raw data gathered from the Smart Ad user study.

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Ergebnisse

Umfrage 396417

Anzahl der Datensätze in dieser Abfrage:	31
Gesamtzahl der Datensätze dieser Umfrage:	31
Anteil in Prozent:	100.00%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für SpielerID

Wie war deine Spieler ID ?

Antwort	Anzahl	Prozent
Antwort	31	100.00%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

ID	Antwort
32	fe33fd2279ac1db3
33	87cb02f46985a755
34	9ff9a3403c900318
35	9774d56d682e549c
40	Tims LG
38	b834e0fb1d4633ea
39	45e528b0edd4a222
41	ac3c44706609cb9f
42	372bc7de0c603061
43	52d887d180ca0572
44	b1ef018e5a2138c6
45	17ea22555cadce32
46	ea2de76ff3642767
47	23e52ba88bf08552
48	9a862b009a2f2cd6
49	9a862b009a2f2cd6-b
52	970af651a0ac4098
53	dee5465b47ae9acb
54	3ff66b6e61cef07a
55	4b88d6eb533ee026
56	73e4d5f5e881b05
57	333ea8c2552b546e
59	8a39f3c6e95fe58e
60	fb32514c9e6ecfe2
61	2060e4dbbbc25eb6
63	2c8d960ac518c267
64	27906f211ecac832
65	2c8d960ac518c267
66	8097201e7f28a347
67	4611d3aaacda3e29
68	fd1aba9f1fd97553

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für AllgPerson [Alter:]

Wie alt bist du?

Berechnung	Ergebnis
Anzahl	31
Summe	913
Standard Abweichung	10.33
Durchschnitt	29.45
Minimum	21
1ter Viertelwert (Q1 unteres Quartil)	25
2ter Viertelwert (Mittleres Quartil)	26
3ter Viertelwert (Q3 Oberes Quartil)	30
Maximum	77

***Null-Werte werden in Berechnungen ausgelassen**
Q1 and Q1 werden berechnet durch die minitab-Methode

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für Allgmein_Beruf

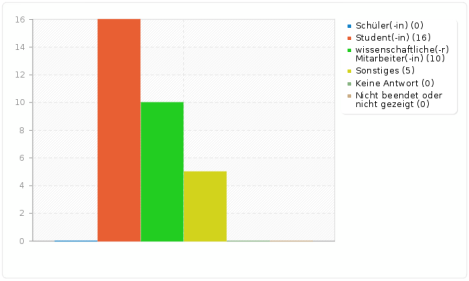
Was übst du für einen Beruf aus?

Antwort	Anzahl	Prozent
Schüler(-in) (A4)	0	0.00%
Student(-in) (A1)	16	51.61%
wissenschaftliche(-r) Mitarbeiter(-in) (A2)	10	32.26%
Sonstiges	5	16.13%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

ID	Antwort
38	Rentner
39	Dipl. Verwaltungswirtin
49	Softwareentwickler
66	BTA

Feld-Zusammenfassung für Allgmein_Beruf

Was übst du für einen Beruf aus?



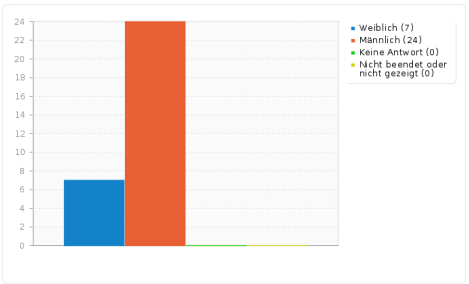
Feld-Zusammenfassung für AllgPersonGeschl

Dein Geschlecht?

Antwort	Anzahl	Prozent
Weiblich (F)	7	22.58%
Männlich (M)	24	77.42%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für AllgPersonGeschl

Dein Geschlecht?



Feld-Zusammenfassung für BASICS_LocalKnowledg

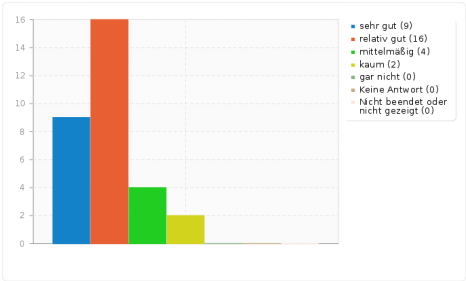
Wie gut kennst du dich in Braunschweig aus?

Antwort	Anzahl	Prozent
sehr gut (A1)	9	29.03%
relativ gut (A2)	16	51.61%
mittelmäßig (A3)	4	12.90%
kaum (A4)	2	6.45%
gar nicht (A5)	0	0.00%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für BASICS_LocalKnowledg

Wie gut kennst du dich in Braunschweig aus?



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für AllgSmartphonenutzun

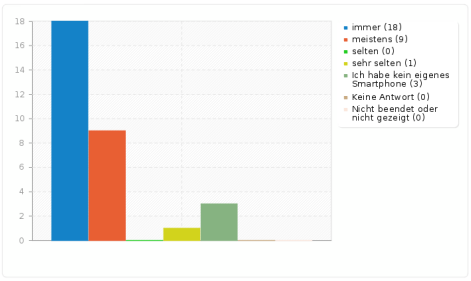
Hast du dein Smartphone / Tablet ständig bei dir?

Antwort	Anzahl	Prozent
immer (A1)	18	58.06%
meistens (A2)	9	29.03%
selten (A4)	0	0.00%
sehr selten (A3)	1	3.23%
Ich habe kein eigenes Smartphone (A5)	3	9.68%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für AllgSmartphonenutzun

Hast du dein Smartphone / Tablet ständig bei dir?



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

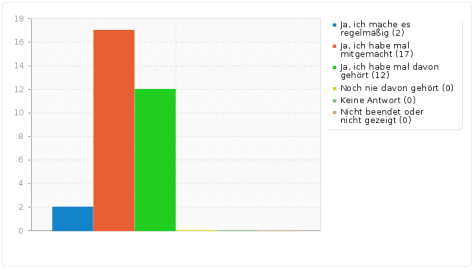
Feld-Zusammenfassung für BASICS_GEOC

Kennst du "Geocaching"?

Antwort	Anzahl	Prozent
Ja, ich mache es regelmäßig (A4)	2	6.45%
Ja, ich habe mal mitgemacht (A3)	17	54.84%
Ja, ich habe mal davon gehört (A2)	12	38.71%
Noch nie davon gehört (A1)	0	0.00%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für BASICS_GEOC

Kennst du "Geocaching"?



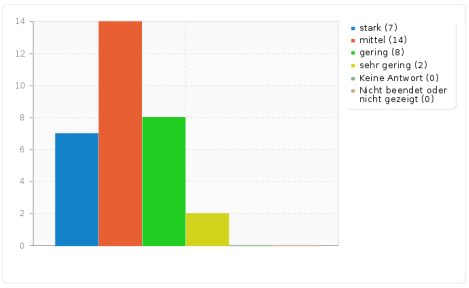
Feld-Zusammenfassung für AllgGeoApps

Wie stark ist allgemein dein Interesse an Apps, die deine Position in die Handlung mit einbeziehen?

Antwort	Anzahl	Prozent
stark (A1)	7	22.58%
mittel (A2)	14	45.16%
gering (A5)	8	25.81%
sehr gering (A4)	2	6.45%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für AllgGeoApps

Wie stark ist allgemein dein Interesse an Apps, die deine Position in die Handlung mit einbeziehen?



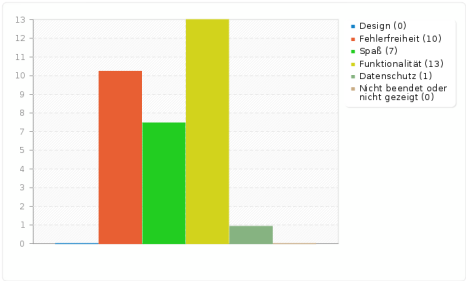
Feld-Zusammenfassung für AkzeptanzEgal [1]

Welche Eigenschaften sind dir allgemein bei Apps wichtig?[Reihenfolge 1]

Antwort	Anzahl	Prozent
Design (A2)	0	0.00%
Fehlerfreiheit (A5)	10	32.26%
Spaß (A4)	7	22.58%
Funktionalität (A3)	13	41.94%
Datenschutz (A1)	1	3.23%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für AkzeptanzEgal [1]

Welche Eigenschaften sind dir allgemein bei Apps wichtig?[Reihenfolge 1]



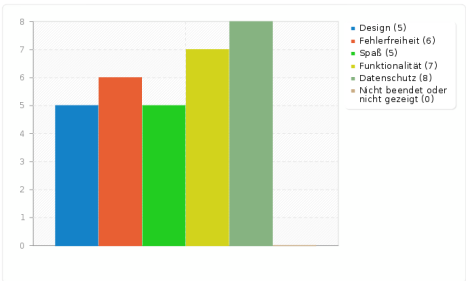
Feld-Zusammenfassung für AkzeptanzEgal [2]

Welche Eigenschaften sind dir allgemein bei Apps wichtig?[Reihenfolge 2]

Antwort	Anzahl	Prozent
Design (A2)	5	16.13%
Fehlerfreiheit (A5)	6	19.35%
Spaß (A4)	5	16.13%
Funktionalität (A3)	7	22.58%
Datenschutz (A1)	8	25.81%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für AkzeptanzEgal [2]

Welche Eigenschaften sind dir allgemein bei Apps wichtig?[Reihenfolge 2]



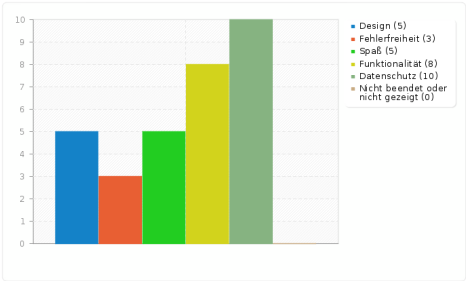
Feld-Zusammenfassung für AkzeptanzEgal [3]

Welche Eigenschaften sind dir allgemein bei Apps wichtig?[Reihenfolge 3]

Antwort	Anzahl	Prozent
Design (A2)	5	16.13%
Fehlerfreiheit (A5)	3	9.68%
Spaß (A4)	5	16.13%
Funktionalität (A3)	8	25.81%
Datenschutz (A1)	10	32.26%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für AkzeptanzEgal [3]

Welche Eigenschaften sind dir allgemein bei Apps wichtig?[Reihenfolge 3]



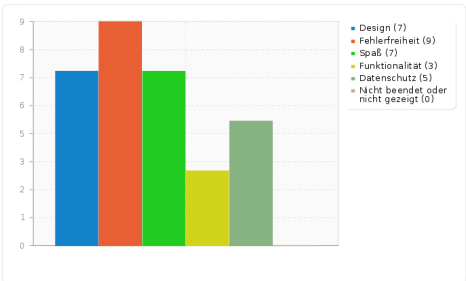
Feld-Zusammenfassung für AkzeptanzEgal [4]

Welche Eigenschaften sind dir allgemein bei Apps wichtig?[Reihenfolge 4]

Antwort	Anzahl	Prozent
Design (A2)	7	22.58%
Fehlerfreiheit (A5)	9	29.03%
Spaß (A4)	7	22.58%
Funktionalität (A3)	3	9.68%
Datenschutz (A1)	5	16.13%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für AkzeptanzEgal [4]

Welche Eigenschaften sind dir allgemein bei Apps wichtig?[Reihenfolge 4]



Feld-Zusammenfassung für AkzeptanzEgal [5]

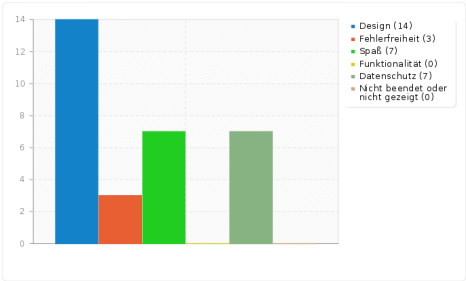
Welche Eigenschaften sind dir allgemein bei Apps wichtig?[Reihenfolge 5]

Antwort	Anzahl	Prozent
Design (A2)	14	45.16%
Fehlerfreiheit (A5)	3	9.68%
Spaß (A4)	7	22.58%
Funktionalität (A3)	0	0.00%
Datenschutz (A1)	7	22.58%
Nicht beendet oder nicht gezeigt	0	0.00%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für AkzeptanzEgal [5]

Welche Eigenschaften sind dir allgemein bei Apps wichtig?[Reihenfolge 5]



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für ANREIZ_weg

Über welchen Weg hast du vom IBR GeoGame erfahren?

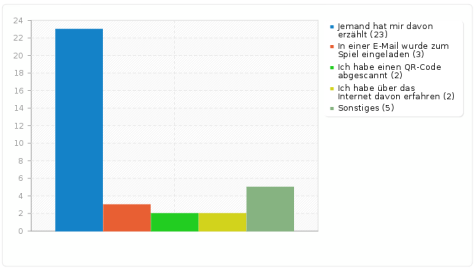
Antwort	Anzahl	Prozent
Jemand hat mir davon erzählt (SQ001)	23	74.19%
In einer E-Mail wurde zum Spiel eingeladen (SQ002)	3	9.68%
Ich habe einen QR-Code abgescannt (SQ003)	2	6.45%
Ich habe über das Internet davon erfahren (SQ004)	2	6.45%
Sonstiges	5	16.13%

ID	Antwort
38	facebook
39	Facebook
54	Ich wurde von meinem Chef gezwungen.
55	Gruppenzwang im IBR;)
56	Flyer in der Mensa

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für ANREIZ_weg

Über welchen Weg hast du vom IBR GeoGame erfahren?



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

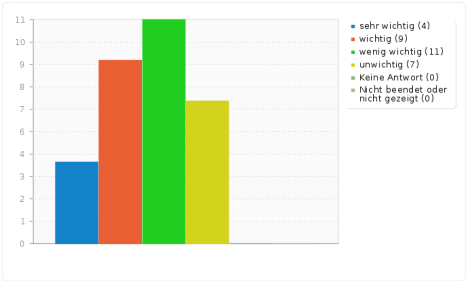
Feld-Zusammenfassung für ANREIZ1(SQ001)

Was war deine Motivation um beim IBR GeoGame mitzumachen und wie wichtig war diese? [Belohnung]

Antwort	Anzahl	Prozent
sehr wichtig (A1)	4	12.90%
wichtig (A2)	9	29.03%
wenig wichtig (A3)	11	35.48%
unwichtig (A4)	7	22.58%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für ANREIZ1(SQ001)

Was war deine Motivation um beim IBR GeoGame mitzumachen und wie wichtig war diese? [Belohnung]



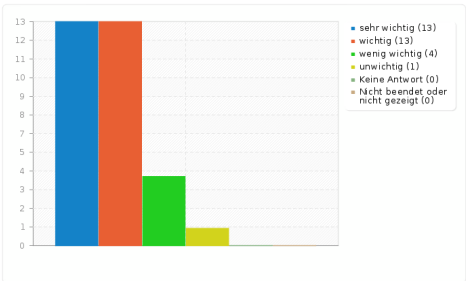
Feld-Zusammenfassung für ANREIZ1(SQ003)

Was war deine Motivation um beim IBR GeoGame mitzumachen und wie wichtig war diese? [Neugier]

Antwort	Anzahl	Prozent
sehr wichtig (A1)	13	41.94%
wichtig (A2)	13	41.94%
wenig wichtig (A3)	4	12.90%
unwichtig (A4)	1	3.23%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für ANREIZ1(SQ003)

Was war deine Motivation um beim IBR GeoGame mitzumachen und wie wichtig war diese? [Neugier]



Feld-Zusammenfassung für ANREIZ1(SQ002)

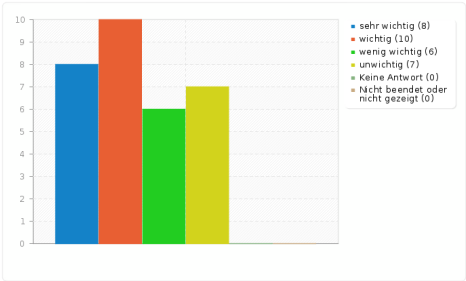
Was war deine Motivation um beim IBR GeoGame mitzumachen und wie wichtig war diese? [Ich wurde von jemandem gebeten]

Antwort	Anzahl	Prozent
sehr wichtig (A1)	8	25.81%
wichtig (A2)	10	32.26%
wenig wichtig (A3)	6	19.35%
unwichtig (A4)	7	22.58%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für ANREIZ1(SQ002)

Was war deine Motivation um beim IBR GeoGame mitzumachen und wie wichtig war diese? [Ich wurde von jemandem gebeten]



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für ANREIZ1(SQ005)

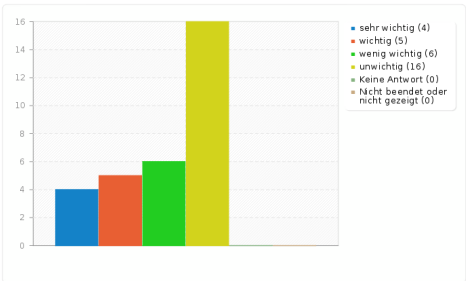
Was war deine Motivation um beim IBR GeoGame mitzumachen und wie wichtig war diese? [Weil mein/e Freund/in auch mitgemacht hat]

Antwort	Anzahl	Prozent
sehr wichtig (A1)	4	12.90%
wichtig (A2)	5	16.13%
wenig wichtig (A3)	6	19.35%
unwichtig (A4)	16	51.61%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für ANREIZ1(SQ005)

Was war deine Motivation um beim IBR GeoGame mitzumachen und wie wichtig war diese? [Weil mein/e Freund/in auch mitgemacht hat]



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für ZIELETransportmittel

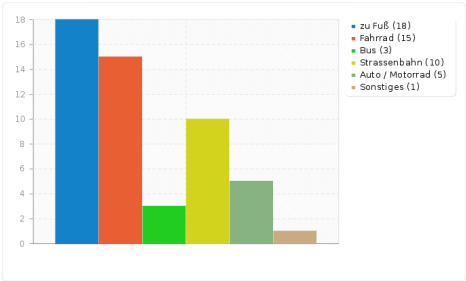
Welche Verkehrsmöglichkeiten hast du zum Erreichen der Ziele in Braunschweig benutzt?

Antwort	Anzahl	Prozent
zu Fuß (SQ001)	18	58.06%
Fahrrad (SQ007)	15	48.39%
Bus (SQ002)	3	9.68%
Strassenbahn (SQ003)	10	32.26%
Auto / Motorrad (SQ005)	5	16.13%
Sonstiges	1	3.23%

ID	Antwort
43	Zug

Feld-Zusammenfassung für ZIELETransportmittel

Welche Verkehrsmöglichkeiten hast du zum Erreichen der Ziele in Braunschweig benutzt?



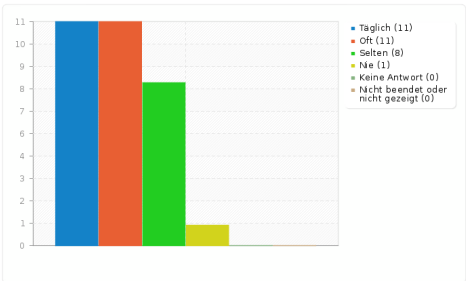
Feld-Zusammenfassung für ZIELEtäglich(SQ004)

Kommst du an einem oder mehreren Zielorten des IBR GeoGame oft vorbei? [Uninähe (Rebenring)]

Antwort	Anzahl	Prozent
Täglich (A1)	11	35.48%
Oft (A2)	11	35.48%
Selten (A3)	8	25.81%
Nie (A4)	1	3.23%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für ZIELEtäglich(SQ004)

Kommst du an einem oder mehreren Zielorten des IBR GeoGame oft vorbei? [Uninähe (Rebenring)]



Feld-Zusammenfassung für ZIELEtäglich(SQ003)

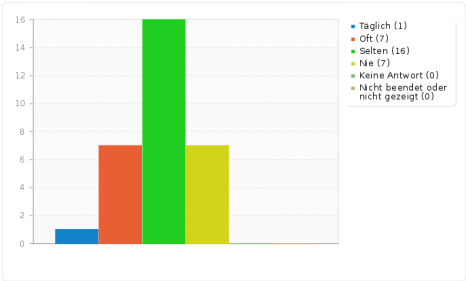
Kommst du an einem oder mehreren Zielorten des IBR GeoGame oft vorbei? [Magniviertel (Magnitor)]

Antwort	Anzahl	Prozent
Täglich (A1)	1	3.23%
Oft (A2)	7	22.58%
Selten (A3)	16	51.61%
Nie (A4)	7	22.58%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für ZIELEtäglich(SQ003)

Kommst du an einem oder mehreren Zielorten des IBR GeoGame oft vorbei? [Magniviertel (Magnitor)]



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für ZIELEtäglich(SQ002)

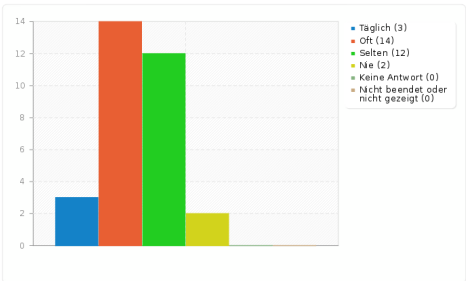
Kommst du an einem oder mehreren Zielorten des IBR GeoGame oft vorbei? [am Stadtkern (Friedrich-Willhelm-Straße)]

Antwort	Anzahl	Prozent
Täglich (A1)	3	9.68%
Oft (A2)	14	45.16%
Selten (A3)	12	38.71%
Nie (A4)	2	6.45%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für ZIELEtäglich(SQ002)

Kommst du an einem oder mehreren Zielorten des IBR GeoGame oft vorbei? [am Stadtkern (Friedrich-Willhelm-Straße)]



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

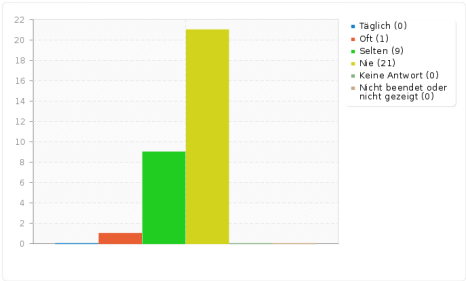
Feld-Zusammenfassung für ZIELEtäglich(SQ005)

Kommst du an einem oder mehreren Zielorten des IBR GeoGame oft vorbei? [Bonuspunkt (Hannover)]

Antwort	Anzahl	Prozent
Täglich (A1)	0	0.00%
Oft (A2)	1	3.23%
Selten (A3)	9	29.03%
Nie (A4)	21	67.74%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für ZIELEtäglich(SQ005)

Kommst du an einem oder mehreren Zielorten des IBR GeoGame oft vorbei? [Bonuspunkt (Hannover)]



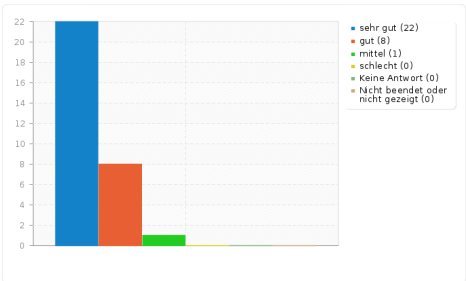
Feld-Zusammenfassung für ZIELE_Schwierigkeit(SQ001)

Wie schwer fandest du beim IBR GeoGame Folgendes: [Erreichbarkeit der Ziele]

Antwort	Anzahl	Prozent
sehr gut (A1)	22	70.97%
gut (A2)	8	25.81%
mittel (A3)	1	3.23%
schlecht (A5)	0	0.00%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für ZIELE_Schwierigkeit(SQ001)

Wie schwer fandest du beim IBR GeoGame Folgendes: [Erreichbarkeit der Ziele]



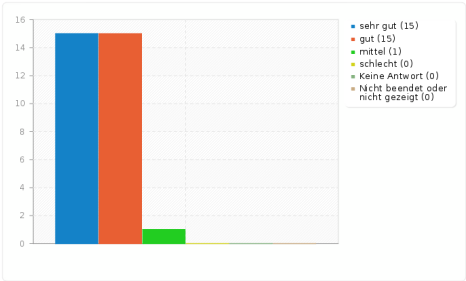
Feld-Zusammenfassung für ZIELE_Schwierigkeit(SQ002)

Wie schwer fandest du beim IBR GeoGame Folgendes: [Bedienung der App]

Antwort	Anzahl	Prozent
sehr gut (A1)	15	48.39%
gut (A2)	15	48.39%
mittel (A3)	1	3.23%
schlecht (A5)	0	0.00%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für ZIELE_Schwierigkeit(SQ002)

Wie schwer fandest du beim IBR GeoGame Folgendes: [Bedienung der App]



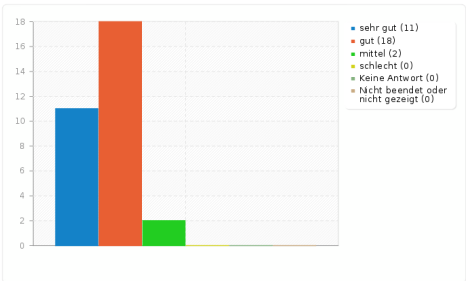
Feld-Zusammenfassung für ZIELE_Schwierigkeit(SQ003)

Wie schwer fandest du beim IBR GeoGame Folgendes: [Funktionalität der App]

Antwort	Anzahl	Prozent
sehr gut (A1)	11	35.48%
gut (A2)	18	58.06%
mittel (A3)	2	6.45%
schlecht (A5)	0	0.00%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für ZIELE_Schwierigkeit(SQ003)

Wie schwer fandest du beim IBR GeoGame Folgendes: [Funktionalität der App]



Feld-Zusammenfassung für Ziele_schwer2

Welche Schwierigkeiten hattest du bei der Erreichbarkeit der Ziele?

Antwort	Anzahl	Prozent
Antwort	1	3.23%
Keine Antwort	1	3.23%
Nicht beendet oder nicht gezeigt	29	93.55%

ID	Antwort
63	GPS war oft sehr stark gestört. Teilweise hat man nur alle paar Minuten eine neue Position empfangen.

Feld-Zusammenfassung für ZIELE_schwer1

Welche Schwierigkeiten hattest du mit der Bedienung der App?

Antwort	Anzahl	Prozent
Antwort	1	3.23%
Keine Antwort	1	3.23%
Nicht beendet oder nicht gezeigt	29	93.55%

ID	Antwort
40	Anwählen mit finger hat nicht immer gleich funktioiert

Feld-Zusammenfassung für ZIELE_schwer3

Welche Schwierigkeiten hattest du mit der Funktionalität der App?

Antwort	Anzahl	Prozent
Antwort	2	6.45%
Keine Antwort	1	3.23%
Nicht beendet oder nicht gezeigt	28	90.32%

ID	Antwort
40	die Karte im Hintergrund ist einige Male verschwunden
63	Die Meldung "evtl. etwas näher zum Empfänger gehen" ist etwas doof, weil man meist nicht weiß wo sich dieser befindet...

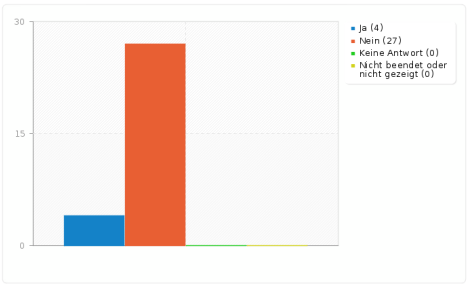
Feld-Zusammenfassung für BonusJaNein

Hast du den Bonuspunkt absolviert?

Antwort	Anzahl	Prozent
Ja (Y)	4	12.90%
Nein (N)	27	87.10%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für BonusJaNein

Hast du den Bonuspunkt absolviert?



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für bonusNeinWarum

Aus welchen Gründen hast du den Bonuspunkt nicht absolviert?

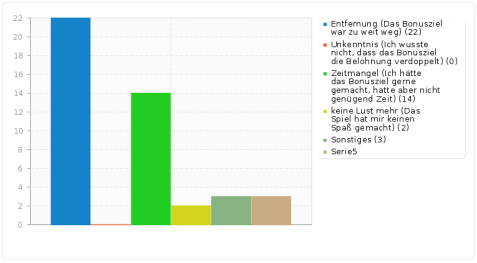
Antwort	Anzahl	Prozent
Entfernung (Das Bonusziel war zu weit weg) (SQ001)	22	70.97%
Unkenntnis (Ich wusste nicht, dass das Bonusziel die Belohnung verdoppelt) (SQ002)	0	0.00%
Zeitmangel (Ich hätte das Bonusziel gerne gemacht, hatte aber nicht genügend Zeit) (SQ003)	14	45.16%
keine Lust mehr (Das Spiel hat mir keinen Spaß gemacht) (SQ004)	2	6.45%
Sonstiges	3	9.68%
Nicht beendet oder nicht gezeigt	3	9.68%

ID	Antwort
45	War gerade vorher in Hannover und wollte nicht schon wieder...
55	Weil ich den blöden Punkt nicht als erstes machen konnte!
60	kein Auto

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für bonusNeinWarum

Aus welchen Gründen hast du den Bonuspunkt nicht absolviert?



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für BONUSwarumJA

Aus welchen Gründen hast du den Bonuspunkt absolviert?

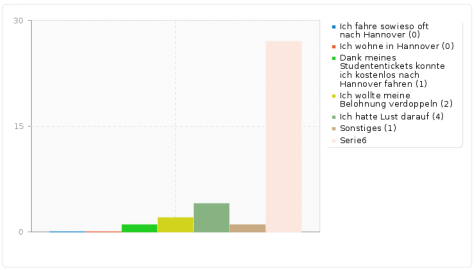
Antwort	Anzahl	Prozent
Ich fahre sowieso oft nach Hannover (SQ004)	0	0.00%
Ich wohne in Hannover (SQ006)	0	0.00%
Dank meines Studententickets konnte ich kostenlos nach Hannover fahren (SQ005)	1	3.23%
Ich wollte meine Belohnung verdoppeln (SQ001)	2	6.45%
Ich hatte Lust darauf (SQ002)	4	12.90%
Sonstiges	1	3.23%
Nicht beendet oder nicht gezeigt	27	87.10%

ID	Antwort
39	Spaß am Finden des Punktes!

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für BONUSwarumJA

Aus welchen Gründen hast du den Bonuspunkt absolviert?



Feld-Zusammenfassung für Verkehrsmöglichkeite

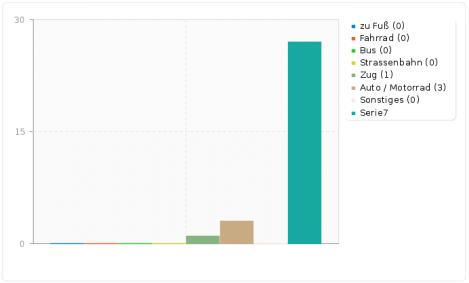
Welche Verkehrsmöglichkeiten hast du zum Erreichen des Bonusziels benutzt?

Antwort	Anzahl	Prozent
zu Fuß (SQ001)	0	0.00%
Fahrrad (SQ007)	0	0.00%
Bus (SQ002)	0	0.00%
Strassenbahn (SQ003)	0	0.00%
Zug (SQ004)	1	3.23%
Auto / Motorrad (SQ005)	3	9.68%
Sonstiges	0	0.00%
Nicht beendet oder nicht gezeigt	27	87.10%

ID	Antwort

Feld-Zusammenfassung für Verkehrsmöglichkeite

Welche Verkehrsmöglichkeiten hast du zum Erreichen des Bonusziels benutzt?



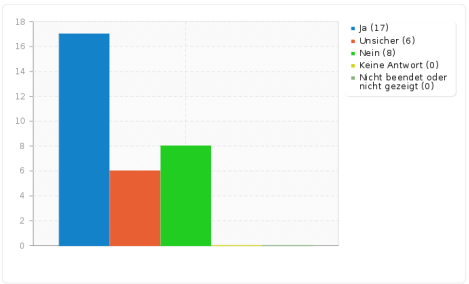
Feld-Zusammenfassung für AkzepSicherheit

"Im Rahmen dieses Spiels wurden keine Bewegungsprofile oder personenbezogenen Daten von dir aufgezeichnet." Glaubst du diese Aussage?

Antwort	Anzahl	Prozent
Ja (A1)	17	54.84%
Unsicher (A2)	6	19.35%
Nein (A3)	8	25.81%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für AkzepSicherheit

"Im Rahmen dieses Spiels wurden keine Bewegungsprofile oder personenbezogenen Daten von dir aufgezeichnet." Glaubst du diese Aussage?



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für AkzepSicherheitJa

Aus welchem Grund glaubst du die Aussage?

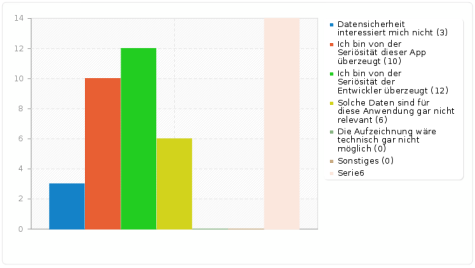
Antwort	Anzahl	Prozent
Datensicherheit interessiert mich nicht (SQ004)	3	9.68%
Ich bin von der Seriösität dieser App überzeugt (SQ002)	10	32.26%
Ich bin von der Seriösität der Entwickler überzeugt (SQ005)	12	38.71%
Solche Daten sind für diese Anwendung gar nicht relevant (SQ001)	6	19.35%
Die Aufzeichnung wäre technisch gar nicht möglich (SQ003)	0	0.00%
Sonstiges	0	0.00%
Nicht beendet oder nicht gezeigt	14	45.16%

ID	Antwort

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für AkzepSicherheitJa

Aus welchem Grund glaubst du die Aussage?



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für AkzepSicherheitNein

Aus welchem Grund glaubst du die Aussage nicht?

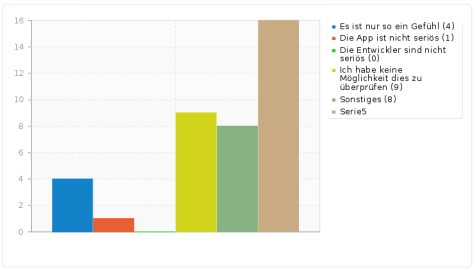
Antwort	Anzahl	Prozent
Es ist nur so ein Gefühl (SQ003)	4	12.90%
Die App ist nicht seriös (SQ004)	1	3.23%
Die Entwickler sind nicht seriös (SQ001)	0	0.00%
Ich habe keine Möglichkeit dies zu überprüfen (SQ002)	9	29.03%
Sonstiges	8	25.81%
Nicht beendet oder nicht gezeigt	16	51.61%

ID	Antwort
40	das ist der sinn der app....
42	Die Daten sind ja nun mal da!
52	weil die App mit googlemaps arbeitet
53	Da es allein zur Funktion der App gehört die verschiedenen Ziele anzulaufen und auch einige Achievements wie "in weniger als 30 Minuten absolviert" integriert sind, warum sollte nicht wenigstens der Weg gespeichert werden? Welchen Sinn hat die App sonst? Da ich keine personenbezogenen Daten angegeben habe und die App auch keine Berechtigungen hat diese auszulesen glaube ich nicht das personenbezogene Daten erfasst wurden.
55	Irgendwer speichert das eh (Google Maps)
57	ist doch klar, dass der entwickler die daten erheben möchte.
60	Ohne eine gewisse Sicherheit kann einem ja alles erzählt werden
61	Zum Erhalt der Belohung musste ich mich in WLAN Access Points anmelden (Ziele), ergo kennen sie zumindest diese drei Positionen

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für AkzepSicherheitNein

Aus welchem Grund glaubst du die Aussage nicht?



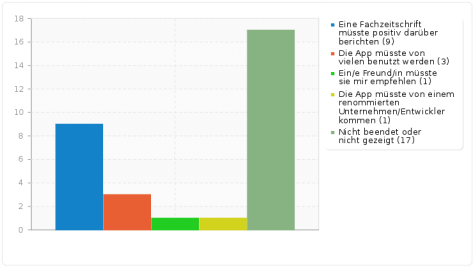
Feld-Zusammenfassung für AkzepVertrauen [1]

Wie sehr würden dich die folgenden Punkte von der Datensicherheit der App überzeugen? [Reihenfolge 1]

Antwort	Anzahl	Prozent
Eine Fachzeitschrift müsste positiv darüber berichten (A1)	9	29.03%
Die App müsste von vielen benutzt werden (A2)	3	9.68%
Ein/e Freund/in müsste sie mir empfehlen (A3)	1	3.23%
Die App müsste von einem renommierten Unternehmen/Entwickler kommen (A4)	1	3.23%
Nicht beendet oder nicht gezeigt	17	54.84%

Feld-Zusammenfassung für AkzepVertrauen [1]

Wie sehr würden dich die folgenden Punkte von der Datensicherheit der App überzeugen? [Reihenfolge 1]



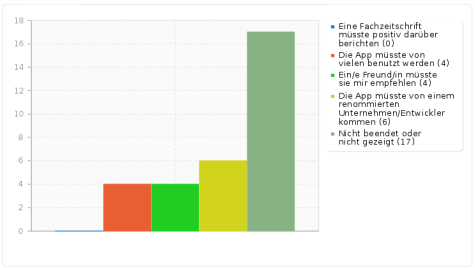
Feld-Zusammenfassung für AkzepVertrauen [2]

Wie sehr würden dich die folgenden Punkte von der Datensicherheit der App überzeugen? [Reihenfolge 2]

Antwort	Anzahl	Prozent
Eine Fachzeitschrift müsste positiv darüber berichten (A1)	0	0.00%
Die App müsste von vielen benutzt werden (A2)	4	12.90%
Ein/e Freund/in müsste sie mir empfehlen (A3)	4	12.90%
Die App müsste von einem renommierten Unternehmen/Entwickler kommen (A4)	6	19.35%
Nicht beendet oder nicht gezeigt	17	54.84%

Feld-Zusammenfassung für AkzepVertrauen [2]

Wie sehr würden dich die folgenden Punkte von der Datensicherheit der App überzeugen? [Reihenfolge 2]



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für AkzepVertrauen [3]

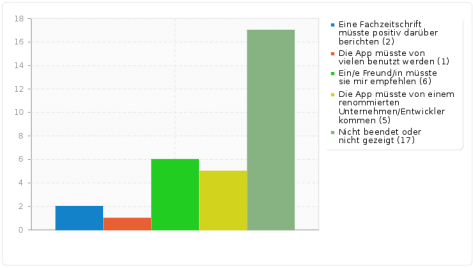
Wie sehr würden dich die folgenden Punkte von der Datensicherheit der App überzeugen? [Reihenfolge 3]

Antwort	Anzahl	Prozent
Eine Fachzeitschrift müsste positiv darüber berichten (A1)	2	6.45%
Die App müsste von vielen benutzt werden (A2)	1	3.23%
Ein/e Freund/in müsste sie mir empfehlen (A3)	6	19.35%
Die App müsste von einem renommierten Unternehmen/Entwickler kommen (A4)	5	16.13%
Nicht beendet oder nicht gezeigt	17	54.84%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für AkzepVertrauen [3]

Wie sehr würden dich die folgenden Punkte von der Datensicherheit der App überzeugen? [Reihenfolge 3]



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für AkzepVertrauen [4]

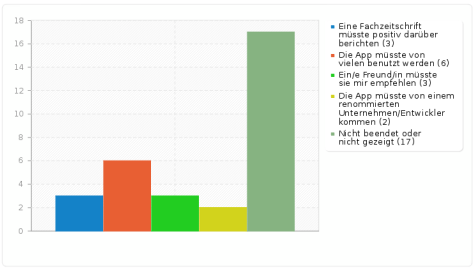
Wie sehr würden dich die folgenden Punkte von der Datensicherheit der App überzeugen? [Reihenfolge 4]

Antwort	Anzahl	Prozent
Eine Fachzeitschrift müsste positiv darüber berichten (A1)	3	9.68%
Die App müsste von vielen benutzt werden (A2)	6	19.35%
Ein/e Freund/in müsste sie mir empfehlen (A3)	3	9.68%
Die App müsste von einem renommierten Unternehmen/Entwickler kommen (A4)	2	6.45%
Nicht beendet oder nicht gezeigt	17	54.84%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für AkzepVertrauen [4]

Wie sehr würden dich die folgenden Punkte von der Datensicherheit der App überzeugen? [Reihenfolge 4]



Feld-Zusammenfassung für AppBerechtigungen

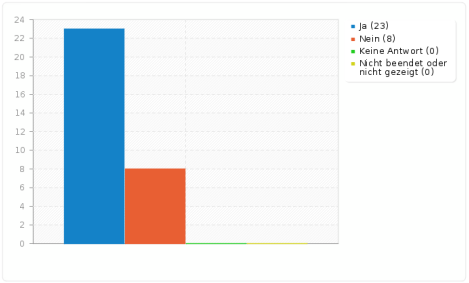
Im App-Beschreibungstext im PlayStore wird erklärt, wozu die GeoGame-App welche Berechtigungen benötigt. Ist dies für dich von Interesse?

Antwort	Anzahl	Prozent
Ja (Y)	23	74.19%
Nein (N)	8	25.81%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für AppBerechtigungen

Im App-Beschreibungstext im PlayStore wird erklärt, wozu die GeoGame-App welche Berechtigungen benötigt. Ist dies für dich von Interesse?



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für Sweets

Wieviele Süßigkeiten hast Du dir ausgesucht?

Berechnung	Ergebnis
Anzahl	31
Summe	1911.0000000000
Standard Abweichung	47.08
Durchschnitt	61.65
Minimum	0.0000000000
1ter Viertelwert (Q1 unteres Quartil)	24
2ter Viertelwert (Mittleres Quartil)	39
3ter Viertelwert (Q3 Oberes Quartil)	87
Maximum	201.0000000000

***Null-Werte werden in Berechnungen ausgelassen**
Q1 and Q1 werden berechnet durch die minitab-Methode

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

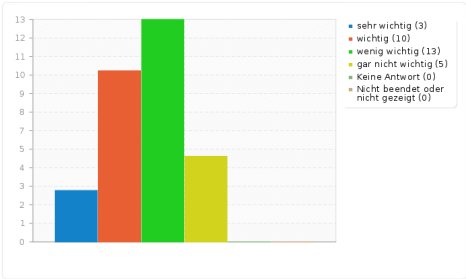
Feld-Zusammenfassung für wichtig Belohnung

Wie wichtig ist dir die Belohnung beim IBR GeoGame?

Antwort	Anzahl	Prozent
sehr wichtig (A1)	3	9.68%
wichtig (A2)	10	32.26%
wenig wichtig (A3)	13	41.94%
gar nicht wichtig (A4)	5	16.13%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für wichtig Belohnung

Wie wichtig ist dir die Belohnung beim IBR GeoGame?



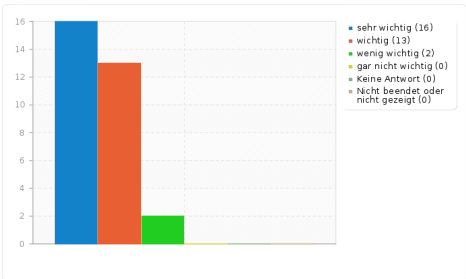
Feld-Zusammenfassung für komplett

Wie wichtig war es dir, alle Ziele zu erreichen?

Antwort	Anzahl	Prozent
sehr wichtig (A1)	16	51.61%
wichtig (A2)	13	41.94%
wenig wichtig (A3)	2	6.45%
gar nicht wichtig (A4)	0	0.00%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für komplett

Wie wichtig war es dir, alle Ziele zu erreichen?



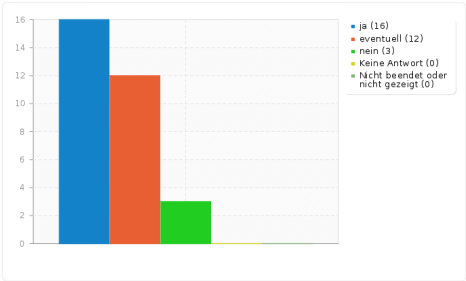
Feld-Zusammenfassung für nochmal

Würdest du das IBR GeoGame nochmals spielen?

Antwort	Anzahl	Prozent
ja (A4)	16	51.61%
eventuell (A5)	12	38.71%
nein (A1)	3	9.68%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für nochmal

Würdest du das IBR GeoGame nochmals spielen?



Feld-Zusammenfassung für nochmalAnreize

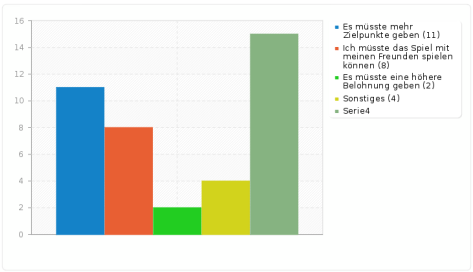
Welche Anreize müssten für dich gegeben sein, damit du das IBR GeoGame nochmal spielen würdest?

Antwort	Anzahl	Prozent
Es müsste mehr Zielpunkte geben (SQ001)	11	35.48%
Ich müsste das Spiel mit meinen Freunden spielen können (SQ002)	8	25.81%
Es müsste eine höhere Belohnung geben (SQ003)	2	6.45%
Sonstiges	4	12.90%
Nicht beendet oder nicht gezeigt	15	48.39%

ID	Antwort
35	für die App-Entwickler und die Forschungsarbeiten müsste es hilfreich sein
49	Erfolge im Spielerprofil
53	Es müsste einen tieferen Sinn haben
63	Die Anwendung müsste interaktiver/unterhaltsamer sein (evtl. eine Art Sightseeing App o.ä)

Feld-Zusammenfassung für nochmalAnreize

Welche Anreize müssten für dich gegeben sein, damit du das IBR GeoGame nochmal spielen würdest?



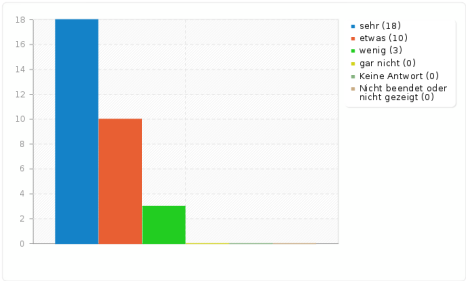
Feld-Zusammenfassung für FreudeÜber(SQ002)

Wie sehr freust du dich über Folgendes: [Die erfolgreiche Erledigung der Aufgabe]

Antwort	Anzahl	Prozent
sehr (A2)	18	58.06%
etwas (A3)	10	32.26%
wenig (A4)	3	9.68%
gar nicht (A6)	0	0.00%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für FreudeÜber(SQ002)

Wie sehr freust du dich über Folgendes: [Die erfolgreiche Erledigung der Aufgabe]



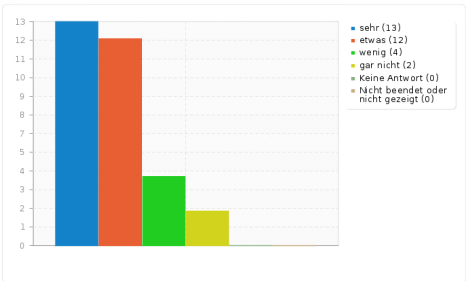
Feld-Zusammenfassung für FreudeÜber(SQ001)

Wie sehr freust du dich über Folgendes: [Die Höhe der Belohnung]

Antwort	Anzahl	Prozent
sehr (A2)	13	41.94%
etwas (A3)	12	38.71%
wenig (A4)	4	12.90%
gar nicht (A6)	2	6.45%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für FreudeÜber(SQ001)

Wie sehr freust du dich über Folgendes: [Die Höhe der Belohnung]



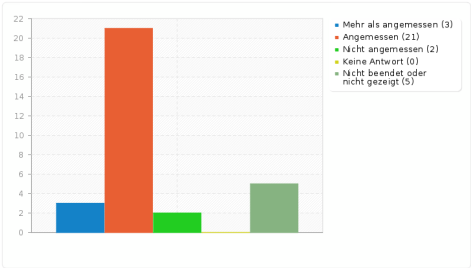
Feld-Zusammenfassung für angemessen

Empfindest Du deine Belohnung als angemessen, im Verhältnis zu deinem Aufwand?

Antwort	Anzahl	Prozent
Mehr als angemessen (A1)	3	9.68%
Angemessen (A5)	21	67.74%
Nicht angemessen (A2)	2	6.45%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	5	16.13%

Feld-Zusammenfassung für angemessen

Empfindest Du deine Belohnung als angemessen, im Verhältnis zu deinem Aufwand?



Feld-Zusammenfassung für artBelohnung

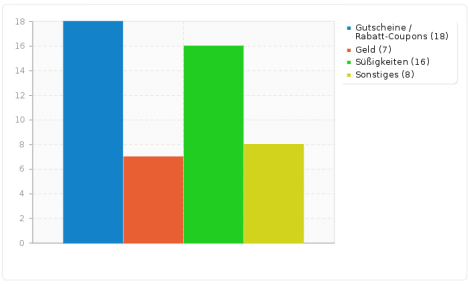
Welche Art von Belohnung würdest Du dir zukünftig wünschen?

Antwort	Anzahl	Prozent
Gutscheine / Rabatt-Coupons (SQ001)	18	58.06%
Geld (SQ002)	7	22.58%
Süßigkeiten (SQ003)	16	51.61%
Sonstiges	8	25.81%

ID	Antwort
35	nicht wichtig
41	Spaß, Rangliste, Community
42	Obst
45	Weltherrschaft
47	Leistungspunkte ;)
49	Spielhandlung (Entertainment)
60	achievements, "digitale abzeichen",
63	Belohnung ist mir nicht wichtig

Feld-Zusammenfassung für artBelohnung

Welche Art von Belohnung würdest Du dir zukünftig wünschen?



Feld-Zusammenfassung für Verlosung

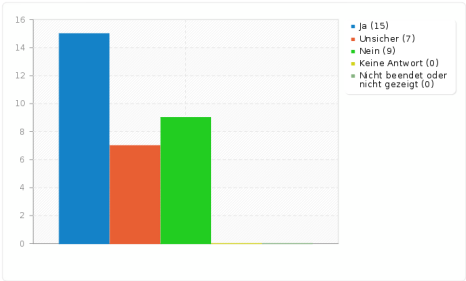
Würdest du auch am IBR GeoGame teilnehmen, wenn es anstelle einer sicheren Belohnung die Möglichkeit einer höheren Belohnung durch eine Verlosung geben würde?

Antwort	Anzahl	Prozent
Ja (A1)	15	48.39%
Unsicher (A2)	7	22.58%
Nein (A3)	9	29.03%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für Verlosung

Würdest du auch am IBR GeoGame teilnehmen, wenn es anstelle einer sicheren Belohnung die Möglichkeit einer höheren Belohnung durch eine Verlosung geben würde?



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für VerlosungArtikel

Was wäre für dich persönlich eine erstrebenswerte Belohnung?

Antwort	Anzahl	Prozent
Antwort	15	48.39%
Keine Antwort	7	22.58%
Nicht beendet oder nicht gezeigt	9	29.03%

ID	Antwort
34	Tolle Geschenke wie Kaffeeautomat, Zelt, ... oder bekocht werden.
38	Essensgutschein
42	Elektronik
43	Erwähnung auf einer Webseite
44	Amazon-Gutschein
46	Gutscheine (Kaffee oder so)
48	Kinokarten, 15 € Gutschein
49	Wenn das Spiel zu einem Gemeinschaftserlebnis wird (Spass)
53	Da für mich eine Belohnung eher zweitrangig ist erstrebe ich auch keine andere Belohnung
54	Merchandisingartikel, die ich nicht im Laden kaufen kann
55	Ein Raspberry Pi, z.B.
60	Gutscheine wären schon gut aber praktikabler wären wohl Errungenschaften in digitaler Form die man über ein spezielles soziales Netzwerk mit anderen Spielern teilen könnte
61	100
67	Einen INGA-Sensorknoten
68	Eine Kleinigkeit, die es aber auf jeden Fall gibt.

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für motivationMatrix(SQ001)

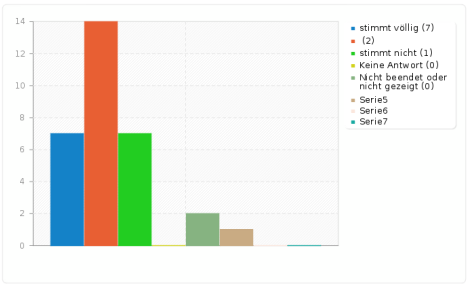
Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil ich denke, dass das Spiel interessant ist]

Antwort	Anzahl	Prozent
stimmt völlig (A1)	7	22.58%
(A2)	14	45.16%
(A3)	7	22.58%
(A4)	0	0.00%
(A5)	2	6.45%
stimmt nicht (A6)	1	3.23%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für motivationMatrix(SQ001)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil ich denke, dass das Spiel interessant ist]



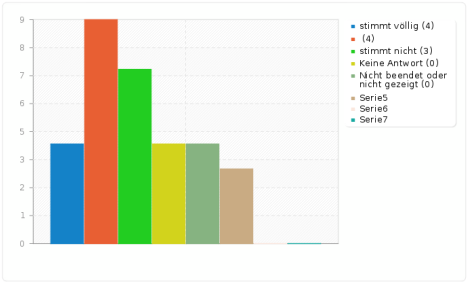
Feld-Zusammenfassung für motivationMatrix(SQ002)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil ich es für mich selbst mache]

Antwort	Anzahl	Prozent
stimmt völlig (A1)	4	12.90%
(A2)	9	29.03%
(A3)	7	22.58%
(A4)	4	12.90%
(A5)	4	12.90%
stimmt nicht (A6)	3	9.68%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für motivationMatrix(SQ002)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil ich es für mich selbst mache]



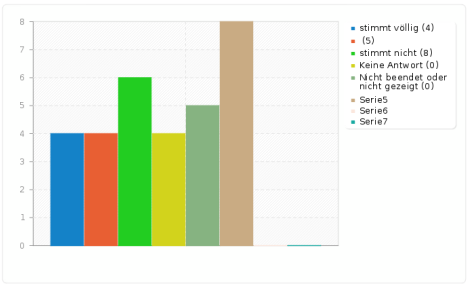
Feld-Zusammenfassung für motivationMatrix(SQ003)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil es von mir erwartet wird]

Antwort	Anzahl	Prozent
stimmt völlig (A1)	4	12.90%
(A2)	4	12.90%
(A3)	6	19.35%
(A4)	4	12.90%
(A5)	5	16.13%
stimmt nicht (A6)	8	25.81%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für motivationMatrix(SQ003)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil es von mir erwartet wird]



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für motivationMatrix(SQ004)

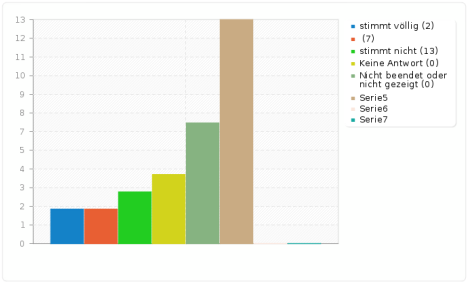
Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [Es gibt sicherlich gute Gründe für das Spiel, aber ich sehe diese nicht]

Antwort	Anzahl	Prozent
stimmt völlig (A1)	2	6.45%
(A2)	2	6.45%
(A3)	3	9.68%
(A4)	4	12.90%
(A5)	7	22.58%
stimmt nicht (A6)	13	41.94%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für motivationMatrix(SQ004)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [Es gibt sicherlich gute Gründe für das Spiel, aber ich sehe diese nicht]



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für motivationMatrix(SQ006)

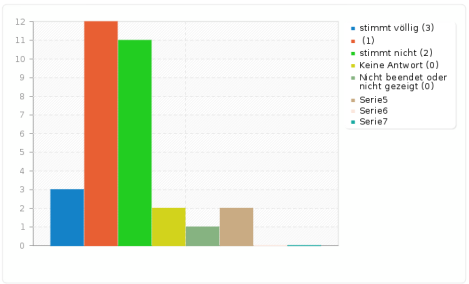
Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil ich denke, das Spiel ist angenehm]

Antwort	Anzahl	Prozent
stimmt völlig (A1)	3	9.68%
(A2)	12	38.71%
(A3)	11	35.48%
(A4)	2	6.45%
(A5)	1	3.23%
stimmt nicht (A6)	2	6.45%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für motivationMatrix(SQ006)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil ich denke, das Spiel ist angenehm]



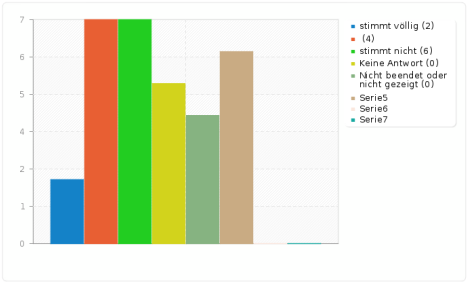
Feld-Zusammenfassung für motivationMatrix(SQ005)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil ich denke, das Spiel ist gut für mich]

Antwort	Anzahl	Prozent
stimmt völlig (A1)	2	6.45%
(A2)	7	22.58%
(A3)	7	22.58%
(A4)	5	16.13%
(A5)	4	12.90%
stimmt nicht (A6)	6	19.35%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für motivationMatrix(SQ005)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil ich denke, das Spiel ist gut für mich]



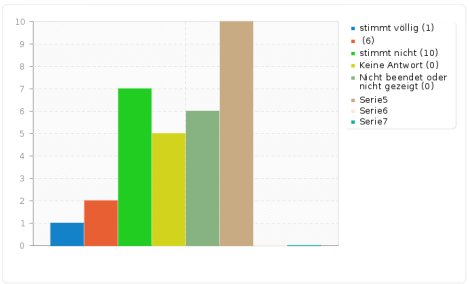
Feld-Zusammenfassung für motivationMatrix(SQ012)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil es etwas ist, was ich zu tun habe]

Antwort	Anzahl	Prozent
stimmt völlig (A1)	1	3.23%
(A2)	2	6.45%
(A3)	7	22.58%
(A4)	5	16.13%
(A5)	6	19.35%
stimmt nicht (A6)	10	32.26%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für motivationMatrix(SQ012)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil es etwas ist, was ich zu tun habe]



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für motivationMatrix(SQ013)

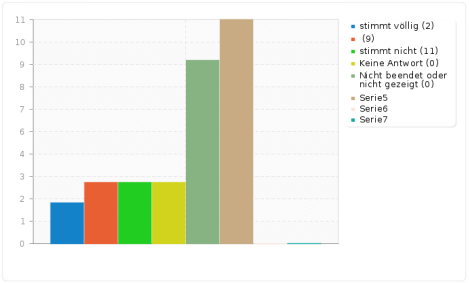
Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [Ich spiele das Spiel, bin mir aber nicht sicher, ob es das wert ist]

Antwort	Anzahl	Prozent
stimmt völlig (A1)	2	6.45%
(A2)	3	9.68%
(A3)	3	9.68%
(A4)	3	9.68%
(A5)	9	29.03%
stimmt nicht (A6)	11	35.48%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für motivationMatrix(SQ013)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [Ich spiele das Spiel, bin mir aber nicht sicher, ob es das wert ist]



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für motivationMatrix(SQ007)

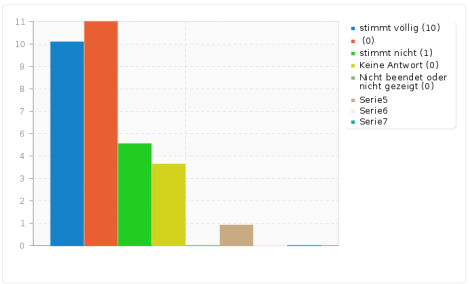
Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil es einfach Spaß macht]

Antwort	Anzahl	Prozent
stimmt völlig (A1)	10	32.26%
(A2)	11	35.48%
(A3)	5	16.13%
(A4)	4	12.90%
(A5)	0	0.00%
stimmt nicht (A6)	1	3.23%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für motivationMatrix(SQ007)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil es einfach Spaß macht]



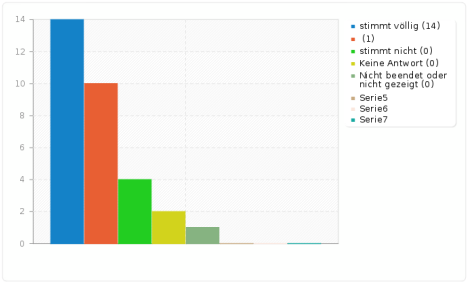
Feld-Zusammenfassung für motivationMatrix(SQ008)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil es meine persönliche Entscheidung war]

Antwort	Anzahl	Prozent
stimmt völlig (A1)	14	45.16%
(A2)	10	32.26%
(A3)	4	12.90%
(A4)	2	6.45%
(A5)	1	3.23%
stimmt nicht (A6)	0	0.00%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für motivationMatrix(SQ008)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil es meine persönliche Entscheidung war]



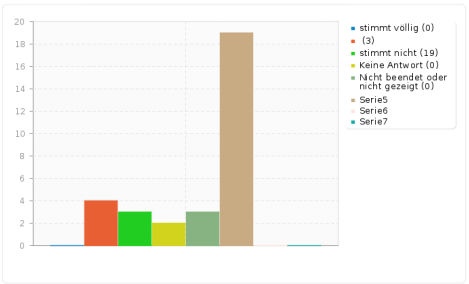
Feld-Zusammenfassung für motivationMatrix(SQ010)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil ich keine Wahl hatte]

Antwort	Anzahl	Prozent
stimmt völlig (A1)	0	0.00%
(A2)	4	12.90%
(A3)	3	9.68%
(A4)	2	6.45%
(A5)	3	9.68%
stimmt nicht (A6)	19	61.29%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für motivationMatrix(SQ010)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil ich keine Wahl hatte]



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für motivationMatrix(SQ014)

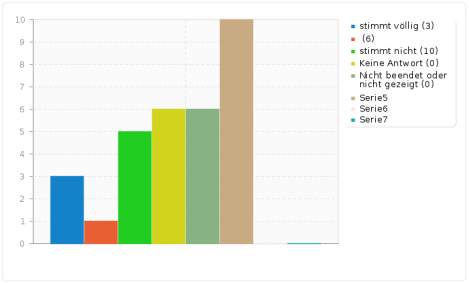
Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [Ich weiß nicht, was das Spiel mir bringt]

Antwort	Anzahl	Prozent
stimmt völlig (A1)	3	9.68%
(A2)	1	3.23%
(A3)	5	16.13%
(A4)	6	19.35%
(A5)	6	19.35%
stimmt nicht (A6)	10	32.26%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für motivationMatrix(SQ014)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [Ich weiß nicht, was das Spiel mir bringt]



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für motivationMatrix(SQ009)

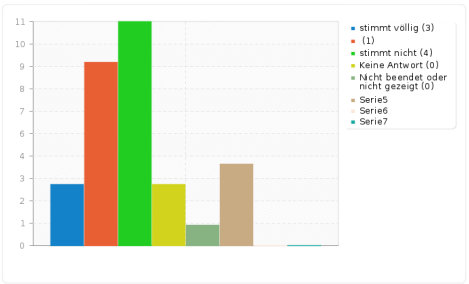
Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil ich mich gut fühle, wenn ich das Spiel spiele]

Antwort	Anzahl	Prozent
stimmt völlig (A1)	3	9.68%
(A2)	9	29.03%
(A3)	11	35.48%
(A4)	3	9.68%
(A5)	1	3.23%
stimmt nicht (A6)	4	12.90%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für motivationMatrix(SQ009)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil ich mich gut fühle, wenn ich das Spiel spiele]



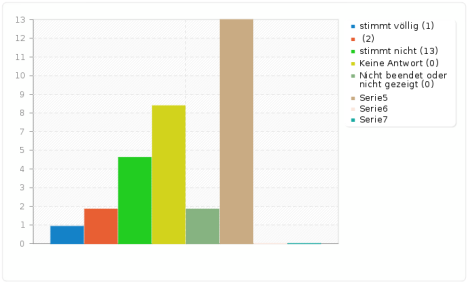
Feld-Zusammenfassung für motivationMatrix(SQ011)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil ich glaube das Spiel ist wichtig für mich]

Antwort	Anzahl	Prozent
stimmt völlig (A1)	1	3.23%
(A2)	2	6.45%
(A3)	5	16.13%
(A4)	8	25.81%
(A5)	2	6.45%
stimmt nicht (A6)	13	41.94%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für motivationMatrix(SQ011)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil ich glaube das Spiel ist wichtig für mich]



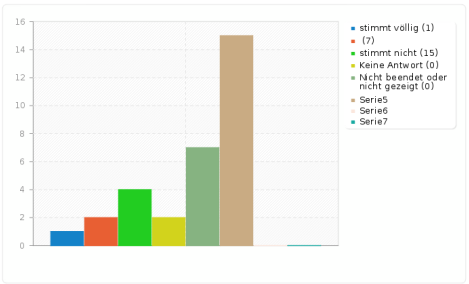
Feld-Zusammenfassung für motivationMatrix(SQ015)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil ich fühle, dass ich es zu tun habe]

Antwort	Anzahl	Prozent
stimmt völlig (A1)	1	3.23%
(A2)	2	6.45%
(A3)	4	12.90%
(A4)	2	6.45%
(A5)	7	22.58%
stimmt nicht (A6)	15	48.39%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Feld-Zusammenfassung für motivationMatrix(SQ015)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [... weil ich fühle, dass ich es zu tun habe]



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für motivationMatrix(SQ016)

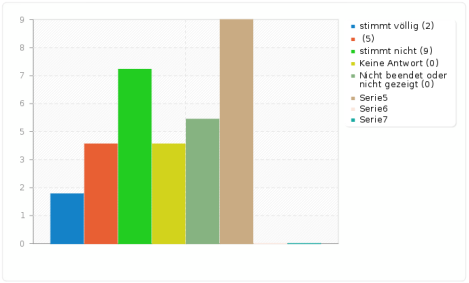
Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [Ich spiele das Spiel, bin mir aber nicht sicher ob ich es sinnvoll ist es nochmal zu spielen]

Antwort	Anzahl	Prozent
stimmt völlig (A1)	2	6.45%
(A2)	4	12.90%
(A3)	7	22.58%
(A4)	4	12.90%
(A5)	5	16.13%
stimmt nicht (A6)	9	29.03%
Keine Antwort	0	0.00%
Nicht beendet oder nicht gezeigt	0	0.00%

Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für motivationMatrix(SQ016)

Im letzten Teil dieser Umfrage bitten wir dich um eine Bewertung der Folgenden Aussagen. Bewerte die Aussagen bitte spontan und ohne lange darüber nachzudenken. Ich habe am Spiel teilgenommen, ... [Ich spiele das Spiel, bin mir aber nicht sicher ob ich es sinnvoll ist es nochmal zu spielen]



Kurz-Statistiken
Umfrage 396417 'TU Braunschweig IBR GeoGame Spielerumfrage'

Feld-Zusammenfassung für Allg_Kommentar

Möchtest du den Entwicklern des IBR Spiels noch etwas mitteilen?

Antwort	Anzahl	Prozent
Antwort	15	48.39%
Keine Antwort	16	51.61%
Nicht beendet oder nicht gezeigt	0	0.00%

ID	Antwort
32	Bei Handys, die lange auf ein GPS-Signal warten müssen, wäre eine Funktion "Ich bin jetzt am Zielpunkt" gut. Nach Auslösen dieser Funktion sollte die App anfangen zu suchen, unabhängig vom GPS/WLAN/Mobilfunk-Standort
34	War echt witzig :)
40	wenn das Handy nicht geliehen wäre hätte ich im Magniviertel abgebrochen, weil ich zweimal am ziel vorbeigefahren bin. Die Distanz zum Ziel habe ich erst nach 20 min verstanden, weil viel zu häufig und zu lange die netzwerksuche aktiv war.
39	Das Spiel hat viel Spaß gemacht. Noch schöner wäre es, wenn das Bonusziel besser mit dem Fahrrad erreichbar wäre!
42	Coole Sache:-)
43	Nettes Spiel. Der Knopf "Karte zentrieren" ist verwirrend. Es ist nicht klar, wann die Funktion eingeschaltet/abgeschaltet ist.
45	Bei jedem Start des App kam die Einführung erneut - das hat etwas gestört...
46	Witzige und gute Idee
47	Ist eine tolle Idee, bitte mehr Orte ;)
48	Bei dem Handy hat das GPS nicht optimal funktioniert, dass war für mich als Spieler etwas frustrierend, da ich den Zielort erreicht hatte. Die Belohnung ist ein sehr starker Anreiz. Ohne ihn hätte ich nicht so viel Spaß gehabt.
49	Good luck!! :)
56	Die App ließ sich nur starten, wenn die GPS-Funktion eingeschaltet war. Ich hätte sie auch gern ohne GPS-Funktion gestartet, um zwischendurch zu gucken, wo das nächste Ziel ist.
57	longclick auf mapfragment ändern mapstyle: gut, aber das sollte auch erwähnt werden (in einem toast), oder ich habe es überlesen...
60	Ihr solltet das Projekt in Studentischen Arbeiten (SEP, Teamprojekt, etc) weiter entwickeln :) Die Integration in ein soziales Netzwerk wäre cool (ob ein eigenes oder ein vorhandenes wie google+ z.B.) Es wäre schön wenn die anwendung viele verschiedene Ziele zum anlaufen hätte und mitspieler eigene Ziele und Gruppen von Zielen in das spiel einbauen können. Bitte behaltet zu jedem ziel auch die kleine Beschreibung bei, die ist nämlich toll :) (könnte twittermäßig auf eine kurze zeichenmenge begrenzt sein) ich bin für gute Benotung des Studenten der dieses Programm geschrieben hat :) btw: insgesammt ist das eine ganz tolle Idee
66	Gute Idee, ausbaufähig. So lernt man eine Stadt gut kennen.

Acronyms

DTN Delay Tolerant Network

BP Bundle Protocol

EID Endpoint Identifier

IPN Interplanetary Network

DHT Distributed Hashtable

BT BitTorrent

MCL Mail Convergence layer

CL Convergence Layer

PBB Primary Bundle Block

TCPCL TCP Convergence Layer

DNS Domain Name Service

WSN Wireless Sensor Network

PDU Protocol Data Unit

SDNV Self-Delimiting Numeric Value

MSB Most Significant Bit

IBF Invertible Bloom Filter

PSN Pocket Switched Network

SDT Self-Determination Theory

URI Uniform Resource Identifier

DNS Domain Name System

RPC Remote Procedure Call

MBMS Multimedia Broadcast Multicast Service

IP Internet Protocol

Bibliography

[1] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet," *ASPLOS-X: Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*, 2002.

[2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-Tolerant Networking Architecture." RFC 4838 (Informational), Apr. 2007.

[3] E. Davies, "DTN - The State of the Art," *N4C*, 2009.

[4] M. V. Barbera, J. Stefa, A. C. Viana, M. D. de Amorim, and M. Boc, "VIP delegation: Enabling vips to offload data in wireless social mobile networks," in *Distributed Computing in Sensor Systems, 7th IEEE International Conference and Workshops, DCOSS 2011, Barcelona, Spain, 27-29 June, 2011, Proceedings*, pp. 1–8, 2011.

[5] K. Scott and S. Burleigh, "Bundle Protocol Specification." RFC 5050 (Experimental), Nov. 2007.

[6] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: rethinking connectivity in developing nations," *Computer*, vol. 37, pp. 78–83, Jan 2004.

[7] A. Galati, T. Bourchas, S. Siby, and S. Mangold, "System architecture for delay tolerant media distribution for rural south africa," in *Proceedings of the 9th ACM international workshop on Wireless network testbeds, experimental evaluation and characterization - WiNTECH '14*, (New York, New York, USA), pp. 65–72, ACM Press, Sept. 2014.

[8] "Cisco Visual Networking Index : Global Mobile Data Traffic Forecast Update , 2012 – 2017," tech. rep., CISCO, 2013.

[9] "The Zettabyte Era: Trends and Analysis," tech. rep., CISCO, 2014.

[10] D. J. W. Andrew S. Tanenbaum, *COMPUTER NETWORKS*. Pearson Education, Inc, 5 ed., 2011.

[11] R. Munroe, *What If?* John Murray Publishers, 2014.

[12] L. Popa, A. Ghodsi, and I. Stoica, "Http as the narrow waist of the future internet," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Hotnets-IX, (New York, NY, USA), pp. 6:1–6:6, ACM, 2010.

[13] Wolf-Bastian Pöttner, Felix Büsching, Georg von Zengen, and Lars Wolf, "Data Elevators: Applying the Bundle Protocol in Delay Tolerant Wireless Sensor Networks," in *The Ninth IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2012)*, (Las Vegas), Oct. 2012.

[14] M. Ramadas, S. Burleigh, and S. Farrell, "Licklider Transmission Protocol - Specification." RFC 5326 (Experimental), Sept. 2008.

[15] S. Schildt, J. Morgenroth, W.-B. Pöttner, and L. Wolf, "IBR-DTN: A lightweight, modular and highly portable Bundle Protocol implementation," *Electronic Communications of the EASST*, vol. 37, pp. 1–11, Jan. 2011.

[16] J. Morgenroth, S. Schildt, and L. Wolf, "A Bundle Protocol Implementation for Android Devices," in *Proceedings of the 18th annual international conference on Mobile computing and networking - Mobicom '12*, (New York, New York, USA), p. 443, ACM Press, Aug. 2012.

[17] M. J. Demmer, J. Ott, and S. Perreault, "Delay Tolerant Networking TCP Convergence Layer Protocol," March 2014.

[18] H. Kruse and S. Ostermann, "UDP Convergence Layers for the DTN Bundle and LTP Protocols," Nov 2008.

[19] H. Kruse, S. Jero, and S. Ostermann, "Datagram Convergence Layers for the Delay- and Disruption-Tolerant Networking (DTN) Bundle Protocol and Licklider Transmission Protocol (LTP)." RFC 7122 (Experimental), Mar. 2014.

[20] "Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," 2006.

[21] T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax." RFC 3986 (INTERNET STANDARD), Jan. 2005. Updated by RFCs 6874, 7320.

[22] W. Eddy and E. Davies, "Using Self-Delimiting Numeric Values in Protocols." RFC 6256 (Informational), May 2011.

[23] S. Schildt, W.-B. Pöttner, O. Ohneiser, and L. Wolf, "NASDI - Naming and Service Discovery for DTNs in Internet Backbones," in *Mobile Wireless Middleware, Operating Systems, and Applications* (C. Borcea, P. Bellavista, C. Giannelli, T. Magedanz, and F. Schreiner, eds.), vol. 65 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 108–121, Springer Berlin Heidelberg, 2013.

[24] S. Schildt, T. Lorentzen, J. Morgenroth, W.-B. Pöttner, and L. Wolf, "Free-riding the BitTorrent DHT to improve DTN connectivity," in *Proceedings of the seventh ACM international workshop on Challenged networks - CHANTS '12*, (New York, New York, USA), p. 9, ACM Press, Aug. 2012.

[25] S. Schildt, B. Gernert, and L. Wolf, "Bundle protocol mail convergence layer: Leveraging legacy internet infrastructure for dtns," in *Proceedings of the 8th ACM MobiCom Workshop on Challenged Networks*, CHANTS '13, (New York, NY, USA), pp. 37–42, ACM, 2013.

[26] P. Hui, A. Chaintreau, R. Gass, J. Scott, J. Crowcroft, and C. Diot, "Pocket switched networking: Challenges, feasibility and implementation issues," in *Autonomic Communication* (I. Stavrakakis and M. Smirnov, eds.), vol. 3854 of *Lecture Notes in Computer Science*, pp. 1–12, Springer Berlin Heidelberg, 2006.

[27] M. Doering, T. Pögel, and L. Wolf, "DTN routing in urban public transport systems," *CHANTS '10: Proceedings of the 5th ACM workshop on Challenged networks*, Sept. 2010.

[28] L. Anders, D. Avri, and S. Olov, "Probabilistic Routing in Intermittently Connected Networks," *SIGMOBILE Mobile Computing and Communication Review*, vol. 7, pp. 19–20, Jan 2004.

[29] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, SenSys '08, (New York, NY, USA), pp. 71–84, ACM, 2008.

[30] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight: won't you be my neighbor?," in *Proceedings of the 18th annual international conference on Mobile computing and networking*, Mobicom '12, (New York, NY, USA), pp. 185–196, ACM, 2012.

[31] J. A. B. Link, C. Wollgarten, S. Schupp, and K. Wehrle, "Perfect difference sets for neighbor discovery: Energy efficient and fair," in *Proceedings of the 3rd Extreme Conference on Communication: The Amazon Expedition*, ExtremeCom '11, (New York, NY, USA), pp. 5:1–5:6, ACM, 2011.

[32] N. Banerjee, M. D. Corner, and B. N. Levine, "Design and Field Experimentation of an Energy-Efficient Architecture for DTN Throwboxes," *IEEE/ACM Transactions on Networking*, vol. 18, pp. 554–567, Apr. 2010.

[33] M. Doering, S. Rottmann, and L. Wolf, "Design and implementation of a low-power energy management module with emergency reserve for solar powered DTN-nodes," in *Proceedings of the 3rd Extreme Conference on Communication The Amazon Expedition - ExtremeCom '11*, (New York, New York, USA), pp. 1–6, ACM Press, Sept. 2011.

[34] D. Ellard and D. Brown, "DTN IP Neighbor Discovery (IPND)," *Internet-Draft*, 2010.

[35] O. P. Waldhorst, "On Overlay-Based Addressing and Routing in Heterogeneous Future Networks," in *Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on*, pp. 1–8, 2010.

[36] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks," Tech. Rep. CS-200006, Duke University, May 2000.

[37] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks," *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pp. 1–11, Apr. 2006.

[38] M. Doering, T. Pögel, and L. C. Wolf, "DTN Routing in Urban Public Transport Systems," in *ACM MobiCom 2010 Workshop on Challenged Networks (CHANTS)*, (Chicago, USA), Sep 2010.

[39] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, Aug 2003.

[40] P. Mockapetris and K. J. Dunlap, "Development of the domain name system," *SIGCOMM Computer Communication Review*, vol. 18, no. 4, pp. 123–133, 1988.

[41] H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, I. Stoica, and M. Walfish, "A Layered Naming Architecture for the Internet," in *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, Aug 2004.

[42] R. Cox, A. Muthitacharoen, and R. Morris, "Serving DNS Using a Peer-to-Peer Lookup Service," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems (IPTPS)*, pp. 155–165, 2002.

[43] J. Ott, "Application protocol design considerations for a mobile internet," in *Proceedings of first ACM/IEEE international workshop on Mobility in the evolving internet architecture (MobiArch)*, 2006.

[44] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications," *IEEE/ACM Transactions on Networking (TON)*, vol. 11, pp. 17–32, Feb 2003.

[45] A. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," in *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg (Middleware)*, Nov 2001.

[46] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, Aug 2001.

[47] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," *Peer-to-Peer Systems*, Jan. 2002.

[48] C. Wang, N. Yang, and H. Chen;, "Improving Lookup Performance Based on Kademlia," in *Proceedings of Second International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC)*, vol. 1, 2010.

[49] Z. Ou, E. Harjula, O. Kassinen, and M. Ylianttila, "Performance evaluation of a Kademlia-based communication-oriented P2P system under churn," *Computer Networks*, vol. 54, pp. 689–705, Apr. 2010.

[50] R. Steinmetz and K. Wehrle, eds., *Peer-to-peer systems and applications*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., Jan 2005.

[51] Delay Tolerant Networking Research Group, "DTN-Bone." http://www.dtnrg.org/wiki/DtnBone, 2013.

[52] S. Symington, S. Farrell, H. Weiss, and P. Lovell, "Bundle Security Protocol Specification." RFC 6257 (Experimental), May 2011.

[53] Andrew Loewenstern, "BEP 5: DHT Protocol." `http://bittorrent.org/beps/bep_0005.html`, 2008.

[54] S. A. Crosby and D. S. Wallach, "An Analysis of Bittorrent's Two Kademlia-based DHTs," *Department of Computer Science, Rice University*, 2007.

[55] D. E. 3rd and P. Jones, "US Secure Hash Algorithm 1 (SHA1)." RFC 3174 (Informational), Sept. 2001. Updated by RFCs 4634, 6234.

[56] Bram Cohen, "BEP 3: The BitTorrent Protocol Specification." `http://bittorrent.org/beps/bep_0003.html`, 2008.

[57] P. Dhungel, D. Wu, B. Schonhorst, and K. W. Ross, "A Measurement Study of Attacks on BitTorrent Leechers," in *Proceedings of the 7th International Conference on Peer-to-peer Systems*, IPTPS'08, (Berkeley, CA, USA), pp. 7–7, USENIX Association, 2008.

[58] Jie Kong, Wandong Cai, Lei Wang, and Qiushi Zhao, "A study of pollution on BitTorrent," in *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, vol. 3, pp. 118–122, IEEE, Feb. 2010.

[59] K. El Defrawy, M. Gjoka, and A. Markopoulou, "Bottorrent: Misusing bittorrent to launch ddos attacks," in *Proceedings of the 3rd USENIX Workshop on Steps to Reducing Unwanted Traffic on the Internet*, SRUTI'07, (Berkeley, CA, USA), pp. 1:1–1:6, USENIX Association, 2007.

[60] J. Postel, "Simple Mail Transfer Protocol." RFC 821 (INTERNET STANDARD), Aug. 1982. Obsoleted by RFC 2821.

[61] J. Klensin, "Simple Mail Transfer Protocol." RFC 5321 (Draft Standard), Oct. 2008.

[62] M. Horton, "UUCP mail interchange format standard." RFC 976, Feb. 1986. Updated by RFC 1137.

[63] B. Kantor and P. Lapsley, "Network News Transfer Protocol." RFC 977 (Proposed Standard), Feb. 1986. Obsoleted by RFC 3977.

[64] C. Feather, "Network News Transfer Protocol (NNTP)." RFC 3977 (Proposed Standard), Oct. 2006. Updated by RFC 6048.

[65] A. Azfar, J. Jiang, L. Shan, M. J. P. Marval, R. Yanggratoke, and S. Ahmed, "ByteWalla : Delay Tolerant Networks on Android phones," tech. rep., KTH Telecommunication Systems Laboratory, 2010.

[66] A. Pentland, R. Fletcher, and A. Hasson, "DakNet: Rethinking Connectivity in Developing Nations," *Computer*, vol. 37, pp. 78–83, Jan. 2004.

[67] P. Guenther and T. Showalter, "Sieve: An Email Filtering Language." RFC 5228 (Proposed Standard), Jan. 2008. Updated by RFCs 5229, 5429, 6785.

[68] B. Gernert and S. Schildt, "Delay Tolerant Networking Email Convergence Layer Protocol," *Internet-Draft*, 2013.

[69] S. Josefsson, "The Base16, Base32, and Base64 Data Encodings." RFC 4648 (Proposed Standard), Oct. 2006.

[70] Google, *Google Peering & Content Delivery: Google Caching Overview* , April 2014.

[71] "A Focus on Efficiency," tech. rep., Facebook, Ericsson, Qualcomm, 2013.

[72] Intel Corporation, "What Happens in an Internet Minute," 2014.

[73] S. Schildt, J. Morgenroth, and L. Wolf, "Efficient false positive free set synchronization using an extended bloom filter approach," *Computer Communications*, vol. 36, no. 10–11, pp. 1245 – 1254, 2013.

[74] A. Broder and M. Mitzenmacher, "Network applications of bloom filters: A survey," *Internet Mathematics*, vol. 1, no. 4, pp. 485–509, 2004.

[75] B. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, Jul 1970.

[76] L. Fan, P. Cao, J. Almeida, and A. Broder, "Summary cache: a scalable wide-area Web cache sharing protocol," *IEEE/ACM Transactions on Networking*, vol. 8, pp. 281 – 293, Jun 2000.

[77] M. Ahmadi and S. Wong, "A cache architecture for counting bloom filters," *15th IEEE International Conference on Networks*, pp. 218–223, 2007.

[78] M. Mitzenmacher, "Compressed Bloom filters," *Networking, IEEE/ACM Transactions on*, vol. 10, no. 5, pp. 604–612, 2002.

[79] P. S. Almeida, C. Baquero, N. Preguica, and D. Hutchison, "Scalable Bloom filters," *Inform. Process. Lett.*, vol. 101, no. 6, pp. 255–261, 2007.

[80] J. Byers, J. Considine, and M. Mitzenmacher, "Fast Approximate Reconciliation of Set Differences," tech. rep., Boston University Computer Science Department, 2002.

[81] J. Byers, J. Considine, M. Mitzenmacher, and S. Rost, "Informed content delivery across adaptive overlay networks," in *Proceedings of the ACM SIGCOMM 2002*, ACM Press, 2002.

[82] Y. Minsky, A. Trachtenberg, and R. Zippel, "Set reconciliation with nearly optimal communication complexity," *IEEE Transactions on Information Theory*, vol. 49, pp. 2213–2218, Sept. 2003.

[83] D. Eppstein, M. T. Goodrich, F. Uyeda, and G. Varghese, "What's the difference?: efficient set reconciliation without prior context," in *Proceedings of the ACM SIGCOMM 2011*, ACM Press, 2011.

[84] M. T. Goodrich and M. Mitzenmacher, "Invertible Bloom Lookup Tables," in *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, pp. 792–799, Sept 2011.

[85] D. Guo and M. Li, "Set Reconciliation via Counting Bloom Filters," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, pp. 2367–2380, Oct 2013.

[86] D. R. Morrison, "PATRICIA—Practical Algorithm To Retrieve Information Coded in Alphanumeric," *Journal of the ACM*, vol. 15, pp. 514–534, Oct. 1968.

[87] R. Merkle, "Method of Providing Digital Signatures." US Patent 4309569, 1982.

[88] VIA Technologies, Inc, "VIA Security Application Note," Tech. Rep. August, VIA Technologies, Inc, aug 2005.

[89] Marvell Technology, *88F6281, 88F6192, and 88F6180 Integrated Controller Functional Specifications*, 2008.

[90] Broadcom Corporation, *BCM5836P Product Brief*, 2007.

[91] S. Schildt and L. Wolf, "Goodies for data: Game-based data propagation in dtns," in *Globecom Workshops (GC Wkshps), 2012 IEEE*, pp. 320–325, Dec 2012.

[92] S. Schildt, T. Lüdtke, K. Reinprecht, and L. Wolf, "User study on the feasibility of incentive systems for smartphone-based dtns in smart cities," in *Proceedings of the 2014 ACM International Workshop on Wireless and Mobile Technologies for Smart Cities*, WiMobCity '14, (New York, NY, USA), pp. 67–76, ACM, 2014.

[93] M. Weiser, "The Computer for the 21st Century," *Scientific American*, vol. 265, pp. 94–104, Sept. 1991.

[94] B. Cohen, "Incentives Build Robustness in BitTorrent," *Workshop on Economics of Peer-to-Peer Systems*, 2003.

[95] D. Kondo, B. Javadi, P. Malecot, F. Cappello, and D. P. Anderson, "Cost-benefit analysis of Cloud Computing versus desktop grids," in *2009 IEEE International Symposium on Parallel & Distributed Processing*, pp. 1–12, May 2009.

[96] J. Su, J. Scott, P. Hui, J. Crowcroft, E. De Lara, C. Diot, A. Goel, M. H. Lim, and E. Upton, "Haggle: Seamless networking for mobile applications," in *Proceedings of the 9th International Conference on Ubiquitous Computing*, UbiComp '07, (Berlin, Heidelberg), pp. 391–408, Springer-Verlag, 2007.

[97] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, pp. 140–150, Sept. 2010.

[98] S. S. Kanhere, "Participatory Sensing: Crowdsourcing Data from Mobile Smartphones in Urban Spaces," *Distributed Computing and Internet Technology*, vol. 7753, pp. 19–26, 2013.

[99] S. Dimatteo, P. Hui, B. Han, and V. O. Li, "Cellular Traffic Offloading through WiFi Networks," in *2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*, pp. 192–201, IEEE, Oct. 2011.

[100] M. V. Barbera, J. Stefa, A. C. Viana, M. D. de Amorim, and M. Boc, "VIP delegation: Enabling VIPs to offload data in wireless social mobile networks," in *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, pp. 1–8, IEEE, June 2011.

[101] U. Shevade and Y. Zhang, "Incentive-aware routing in DTNs," in *2008 IEEE International Conference on Network Protocols*, pp. 238–247, IEEE, Oct. 2008.

[102] A. Garyfalos and K. Almeroth, "Coupons: A Multilevel Incentive Scheme for Information Dissemination in Mobile Networks," *IEEE Transactions on Mobile Computing*, vol. 7, pp. 792–804, June 2008.

[103] R. Lu, X. Lin, H. Zhu, X. Shen, and B. Preiss, "Pi: A practical incentive protocol for delay tolerant networks," *IEEE Transactions on Wireless Communications*, vol. 9, pp. 1483–1493, Apr. 2010.

[104] B. B. Chen and M. C. Chan, "MobiCent: a Credit-Based Incentive System for Disruption Tolerant Network," in *2010 Proceedings IEEE INFOCOM*, pp. 1–9, IEEE, Mar. 2010.

[105] N. Ellison and D. Boyd, "Social network sites: Definition, history, and scholarship," *Journal of Computer-Mediated Communication*, vol. 13, no. 1, pp. 210–230, 2007.

[106] H. Haddadi, P. Hui, and I. Brown, "MobiAd," in *Proceedings of the fifth ACM international workshop on Mobility in the evolving internet architecture - MobiArch '10*, (New York, New York, USA), p. 33, ACM Press, Sept. 2010.

[107] R. E. Nisbett and T. D. Wilson, "Telling more than we can know: Verbal reports on mental processes.," *Psychological Review*, vol. 84, pp. 231–259, 1977.

[108] R. Likert, "A technique for the measurement of attitudes.," *Archives of Psychology*, vol. 22, no. 140, 1932.

[109] BITKOM e.V., "Das Handy als ständiger Begleiter." http://www.bitkom.org/de/presse/30739_77337.aspx, 9 2013.

[110] M. Gidda, "Edward Snowden and the NSA files - timeline," *The Guardian*, July 2013.

[111] R. M. Ryan and E. L. Deci, "Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions," *Contemporary Educational Psychology*, vol. 25, no. 1, pp. 54–67, 2000.

[112] M. Gagné and E. L. Deci, "Self-determination theory and work motivation," *Journal of Organizational Behavior*, vol. 26, pp. 331–362, June 2005.

[113] J. Galtung, "Violence, Peace, and Peace Research," *Journal of Peace Research*, vol. 6, no. 3, pp. 167–191, 1969.

[114] L. Smyth, "The Demographics of Ingress." http://simulacrum.cc/2013/01/23/the-demographics-of-ingress/, January 2013.

[115] P. Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Core." RFC 6120 (Proposed Standard), Mar. 2011.

[116] P. Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence." RFC 6121 (Proposed Standard), Mar. 2011.

[117] D. J. Mudgway, "Telecommunications and Data Acquisition Systems Support for the Viking 1975 Mission to Mars," tech. rep., Jet Propulsion Laboratory, 1983.

[118] D. Powell, "Lasers boost space communications," *Nature*, vol. 499, pp. 266–267, 2013.

[119] D. M. Boroson, B. S. Robinson, D. V. Murphy, D. A. Burianek, F. Khatri, J. M. Kovalik, Z. Sodnik, and D. M. Cornwell, "Overview and results of the Lunar Laser Communication Demonstration," in *SPIE LASE* (H. Hemmati and D. M. Boroson, eds.), p. 89710S, International Society for Optics and Photonics, Mar. 2014.

[120] "Lunar Laser Communication Demonstration NASA's First Space Laser Communication System Demonstration," tech. rep., NASA, 2013.

[121] J. Postel, "Internet Protocol." RFC 791 (INTERNET STANDARD), Sept. 1981. Updated by RFCs 1349, 2474, 6864.

Institute of Operating Systems and Computer Networks - IBR
Sebastian Schildt
Mühlenpfordtstr. 23
38106 Braunschweig, Germany